INSTITUTO POLITECNICO NACIONAL

Escuela Superior de Cómputo

Practica 5: Análisis de sentimientos, polaridad u opinión.

Unidad de aprendizaje: Tecnologías de Lenguaje Natural

Alumno:

Flores Lara Alberto

Profesor: Flores Estrada Ituriel Enrique

Grupo 5BV1

INDICE

INTRODUCCION

En esta práctica de Tecnologías de Lenguaje Natural, llevamos a cabo un análisis de reseñas de productos para identificar y clasificar los sentimientos expresados en los textos. Implementamos y comparamos diferentes modelos y enfoques para evaluar su eficacia en la tarea de clasificación de sentimientos.

OBJETIVO DE LA PRÁCTICA

El objetivo es crear un notebook en jupyter para realizar el análisis de sentimientos en reseñas de productos, utilizando múltiples enfoques y técnicas.

1. **Adquisición de Datos:**
    - Descargar el conjunto de datos de reseñas de Amazon desde Kaggle.

2. **Análisis Exploratorio de Datos (EDA):**
    - Describir el conjunto de datos utilizando comandos de Python para obtener información general, las primeras filas del DataFrame, dimensiones y caracterización de cada columna.

3. **Preprocesamiento:**
    - Justificar y seleccionar las dimensiones necesarias para el análisis.
    - Eliminar las columnas irrelevantes.
    - Convertir las calificaciones en etiquetas de sentimiento (Negativo, Neutral, Positivo).
    - Validar y balancear las clases para asegurar una distribución equitativa de los datos.

4. **Limpieza de Datos:**
    - Analizar las reseñas para identificar y justificar las técnicas de limpieza de texto adecuadas.
    - Implementar funciones para limpiar el texto eliminando etiquetas HTML, correos electrónicos, URLs, signos de puntuación y stopwords.

5. **Transformación / Normalización de Datos:**
    - Convertir todos los términos a minúsculas.
    - Aplicar técnicas de stemming y lematización para normalizar las palabras en los textos.
    - Tokenizar las reseñas.
    - Vectorizar las reseñas utilizando técnicas de TF-IDF y One-Hot Encoding.

6. **Análisis de Sentimientos usando Diccionarios:**
    - Utilizar el diccionario Harvard IV-4 con el módulo pysentiment2.

- Utilizar el Opinion Lexicon con el módulo NLTK.

7. **Análisis de Sentimientos usando Algoritmos de Aprendizaje de Máquina:**
    - Implementar y evaluar modelos de Regresión Logística, Árboles de Decisión y Máquinas de Soporte Vectorial (SVM).
    - Realizar la comparación entre modelos utilizando validación cruzada (k-fold cross-validation).

8. **Análisis de Sentimientos usando Word Embeddings y Redes Neuronales:**
    - Usar una capa de Word embeddings preconstruida.
    - Usar una capa de Word embeddings aprendida a partir del cuerpo de documentos a analizar.
    - Realizar la comparación entre modelos utilizando validación cruzada (k-fold cross-validation).

DESARROLLO PUNTO POR PUNTO DE LA PRACTICA

Adquisición de Datos:

Para esta práctica, utilizamos un conjunto de datos que contiene reseñas de productos de comida fina en Amazon.

Análisis Exploratorio de Datos (EDA):

El conjunto de datos incluye varias columnas que representan diferentes aspectos de las reseñas, tales como el identificador del producto, el identificador del usuario, el texto de la reseña, la puntuación otorgada, entre otros.

```
Informacion general del DataFrame:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 568454 entries, 0 to 568453
Data columns (total 10 columns):
 #   Column                  Non-Null Count   Dtype
---  ------                  --------------   -----
 0   Id                      568454 non-null  int64
 1   ProductId               568454 non-null  object
 2   UserId                  568454 non-null  object
 3   ProfileName             568428 non-null  object
 4   HelpfulnessNumerator    568454 non-null  int64
 5   HelpfulnessDenominator  568454 non-null  int64
 6   Score                   568454 non-null  int64
 7   Time                    568454 non-null  int64
 8   Summary                 568427 non-null  object
 9   Text                    568454 non-null  object
dtypes: int64(5), object(5)
memory usage: 43.4+ MB
None

Primeras filas del DataFrame:
   Id   ProductId          UserId                      ProfileName  \
0   1  B001E4KFG0  A3SGXH7AUHU8GW                       delmartian
1   2  B00813GRG4  A1D87F6ZCVE5NK                           dll pa
2   3  B000LQOCH0   ABXLMWJIXXAIN  Natalia Corres "Natalia Corres"
3   4  B000UA0QIQ  A395BORC6FGVXV                             Karl
4   5  B006K2ZZ7K  A1UQRSCLF8GW1T    Michael D. Bigham "M. Wassir"

   HelpfulnessNumerator  HelpfulnessDenominator  Score        Time  \
0                     1                       1      5  1303862400
1                     0                       0      1  1346976000
2                     1                       1      4  1219017600
3                     3                       3      2  1307923200
4                     0                       0      5  1350777600

                Summary                                                Text
0  Good Quality Dog Food  I have bought several of the Vitality canned d...
1      Not as Advertised  Product arrived labeled as Jumbo Salted Peanut...
2  "Delight" says it all  This is a confection that has been around a fe...
3         Cough Medicine  If you are looking for the secret ingredient i...
4            Great taffy  Great taffy at a great price.  There was a wid...

Dimensiones del DataFrame:
(568454, 10)
```

Resultados del primer punto de la practica

```
Caracterizacion de cada columna:

Columna: Id
Tipo de dato: int64
Descripcion:
count    568454.000000
mean     284227.500000
std      164098.679298
min           1.000000
25%      142114.250000
50%      284227.500000
75%      426340.750000
max      568454.000000
Name: Id, dtype: float64

Columna: ProductId
Tipo de dato: object
Descripcion:
count         568454
unique         74258
top       B007JFMH8M
freq             913
Name: ProductId, dtype: object

Columna: UserId
Tipo de dato: object
Descripcion:
count             568454
unique            256059
top       A3OXHLG6DIBRW8
freq                 448
Name: UserId, dtype: object

Columna: ProfileName
Tipo de dato: object
Descripcion:
count               568428
unique              218415
top       C. F. Hill "CFH"
freq                   451
Name: ProfileName, dtype: object

Columna: HelpfulnessNumerator
Tipo de dato: int64
Descripcion:
count    568454.000000
mean          1.743817
std           7.636513
min           0.000000
25%           0.000000
50%           0.000000
75%           2.000000
max         866.000000
Name: HelpfulnessNumerator, dtype: float64
```

Resultados del primer punto de la practica

```
Columna: HelpfulnessDenominator
Tipo de dato: int64
Descripcion:
count    568454.00000
mean          2.22881
std           8.28974
min           0.00000
25%           0.00000
50%           1.00000
75%           2.00000
max         923.00000
Name: HelpfulnessDenominator, dtype: float64

Columna: Score
Tipo de dato: int64
Descripcion:
count    568454.000000
mean          4.183199
std           1.310436
min           1.000000
25%           4.000000
50%           5.000000
75%           5.000000
max           5.000000
Name: Score, dtype: float64

Columna: Time
Tipo de dato: int64
Descripcion:
count    5.684540e+05
mean     1.296257e+09
std      4.804331e+07
min      9.393408e+08
25%      1.271290e+09
50%      1.311120e+09
75%      1.332720e+09
max      1.351210e+09
Name: Time, dtype: float64

Columna: Summary
Tipo de dato: object
Descripcion:
count         568427
unique        295742
top       Delicious!
freq            2462
Name: Summary, dtype: object

Columna: Text
Tipo de dato: object
Descripcion:
count                                                568454
unique                                               393579
top       This review will make me sound really stupid, ...
freq                                                    199
Name: Text, dtype: object
```

Resultados del primer punto de la practica

- Carga de Datos: Utilizamos pandas para cargar el conjunto de datos desde un archivo CSV.

- Información General: df.info() proporciona una visión general de las columnas del DataFrame, incluyendo el tipo de datos y el número de valores no nulos.

- Primeras Filas: df.head() muestra las primeras filas del DataFrame.

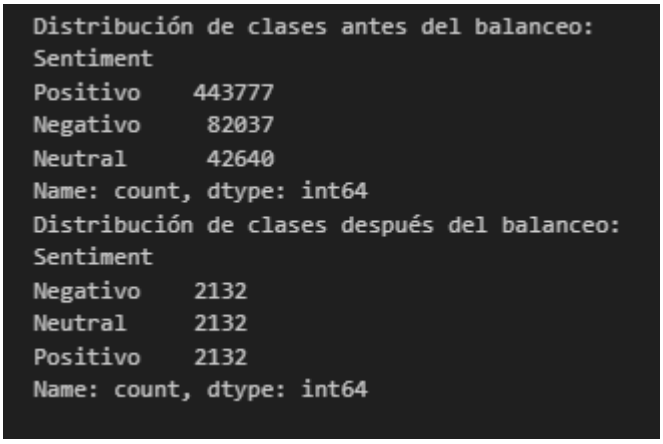
- Dimensiones: df.shape devuelve las dimensiones del DataFrame, mostrando el número de filas y columnas.

- Caracterización de Columnas: Iteramos sobre las columnas del DataFrame para mostrar el tipo de dato y una breve descripción estadística de cada una utilizando df[column].describe().

Preprocesamiento:

En esta etapa, realizamos varias tareas de preprocesamiento para preparar los datos para el análisis de sentimiento. Esto incluye la eliminación de columnas irrelevantes, la conversión de las puntuaciones en categorías de sentimiento, y el balanceo de clases.

Estos fueron los resultados obtenidos:



Resultados del tercer punto de la practica

- Selección de Columnas Relevantes: Mantenemos sólo las columnas que son relevantes para el análisis de sentimiento (Score y Text; ya que Score la utilizaremos para la clasificación de sentimientos y Text es la columna que contienen las reseñas las cuales analizaremos).

- Conversión de Puntuaciones a Sentimientos: Creamos una nueva columna Sentiment que categoriza las puntuaciones en Negativo, Neutral y Positivo usando la función convert_score_to_sentiment (asignamos la etiqueta “Negativo” si el score esta entre 1 y 2; “Positivo” si el score esta entre 4 y 5; y “Neutro si el score es 3).

- Eliminación de la Columna Score: Eliminamos la columna Score ya que ahora tenemos la columna Sentiment.

- Balanceo de Clases: Verificamos la distribución de las clases antes del balanceo. Luego, balanceamos las clases para asegurarnos de que haya un número equitativo de muestras en cada categoría de sentimiento utilizando la función resample de sklearn. Además, cambiamos la cantidad de muestras a una veinteava parte de las muestras obtenidas después del balanceo, para poder hacer un análisis eficiente; en caso de utilizar todas las muestras tardaría poco mas de 10 horas la compilación de resultados.

Limpieza de Datos:

En este punto, aplicamos varias técnicas de limpieza de texto para preparar los datos para su posterior análisis. Esto incluye la eliminación de etiquetas HTML, direcciones de correo electrónico, URLs, signos de puntuación y stopwords.

A continuación, se presentan los resultados obtenidos:

```
Texto original:
 534435    Well, these capsules are sold by Top Line and ...
52570     No seriously, this is ridiculous, $45 for a ba...
92327     I like strong coffee but was able to use the c...
165543    Started out OK, but after a few weeks my dog d...
201034    The taste of raspberry is pretty much all I ge...
Name: Text, dtype: object

Texto limpio:
 534435    well capsules sold top line fulfilled amazon t...
52570     seriously ridiculous bag corn obligate carnivo...
92327     like strong coffee able use cup times reminds ...
165543    started ok weeks dog decided didnt like taste ...
201034    taste raspberry pretty much get taste smell ch...
Name: Cleaned_Text, dtype: object
```

- Función clean_text: Definimos una función para limpiar el texto que:
  - Convierte el texto a minúsculas.
  - Elimina etiquetas HTML utilizando una expresión regular.
  - Elimina direcciones de correo electrónico utilizando una expresión regular.
  - Elimina URLs utilizando una expresión regular.
  - Elimina signos de puntuación utilizando str.translate y string.punctuation.
  - Elimina números utilizando una expresión regular.
  - Elimina stopwords utilizando nltk para tokenizar el texto y filtrar las stopwords.

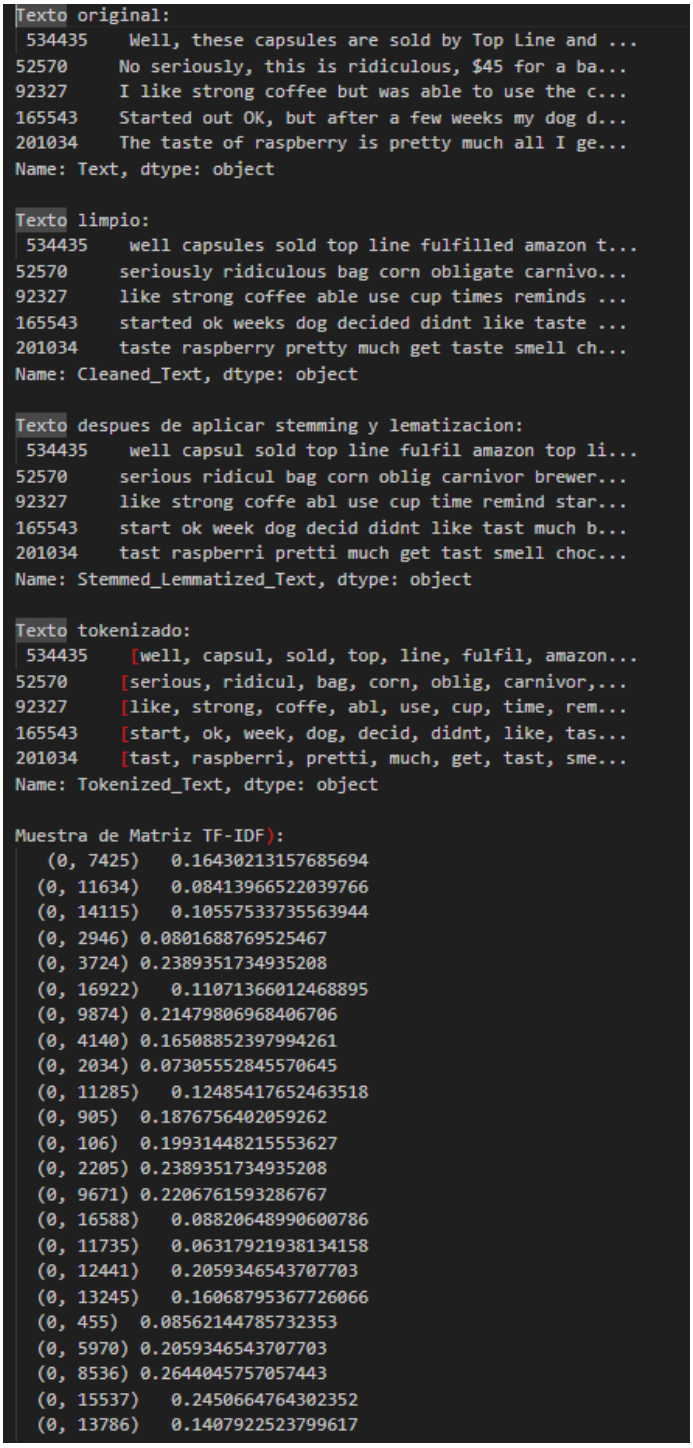
- Aplicar la Función de Limpieza: Aplicamos la función de limpieza al texto de las reseñas en la columna Text y guardamos el resultado en una nueva columna Cleaned_Text.

- Verificación de Resultados: Mostramos las primeras filas del texto original y el texto limpio para verificar la limpieza.

Transformación / Normalización de Datos:

En esta etapa, aplicamos técnicas de transformación y normalización de los datos textuales. Esto incluye convertir todos los términos a minúsculas, realizar stemming y lematización, tokenización y vectorización de las reseñas.

Estos fueron los resultados:



```
Texto original:
 534435    Well, these capsules are sold by Top Line and ...
52570     No seriously, this is ridiculous, $45 for a ba...
92327     I like strong coffee but was able to use the c...
165543    Started out OK, but after a few weeks my dog d...
201034    The taste of raspberry is pretty much all I ge...
Name: Text, dtype: object

Texto limpio:
 534435    well capsules sold top line fulfilled amazon t...
52570     seriously ridiculous bag corn obligate carnivo...
92327     like strong coffee able use cup times reminds ...
165543    started ok weeks dog decided didnt like taste ...
201034    taste raspberry pretty much get taste smell ch...
Name: Cleaned_Text, dtype: object

Texto despues de aplicar stemming y lematizacion:
 534435    well capsul sold top line fulfil amazon top li...
52570     serious ridicul bag corn oblig carnivor brewer...
92327     like strong coffe abl use cup time remind star...
165543    start ok week dog decid didnt like tast much b...
201034    tast raspberri pretti much get tast smell choc...
Name: Stemmed_Lemmatized_Text, dtype: object

Texto tokenizado:
 534435    [well, capsul, sold, top, line, fulfil, amazon...
52570     [serious, ridicul, bag, corn, oblig, carnivor,...
92327     [like, strong, coffe, abl, use, cup, time, rem...
165543    [start, ok, week, dog, decid, didnt, like, tas...
201034    [tast, raspberri, pretti, much, get, tast, sme...
Name: Tokenized_Text, dtype: object

Muestra de Matriz TF-IDF):
   (0, 7425)   0.16430213157685694
  (0, 11634)   0.08413966522039766
  (0, 14115)   0.10557533735563944
  (0, 2946) 0.0801688769525467
  (0, 3724) 0.2389351734935208
  (0, 16922)   0.11071366012468895
  (0, 9874) 0.21479806968406706
  (0, 4140) 0.16508852397994261
  (0, 2034) 0.07305552845570645
  (0, 11285)   0.12485417652463518
  (0, 905)  0.1876756402059262
  (0, 106)  0.19931448215553627
  (0, 2205) 0.2389351734935208
  (0, 9671) 0.2206761593286767
  (0, 16588)   0.08820648990600786
  (0, 11735)   0.06317921938134158
  (0, 12441)   0.2059346543707703
  (0, 13245)   0.16068795367726066
  (0, 455)  0.08562144785732353
  (0, 5970) 0.2059346543707703
  (0, 8536) 0.2644045757057443
  (0, 15537)   0.2450664764302352
  (0, 13786)   0.1407922523799617
```

Resultados del quinto punto de la practica

- Stemming y Lemmatización:

    - Utilizamos PorterStemmer para realizar stemming, que reduce las palabras a sus raíces morfológicas.
    - Utilizamos WordNetLemmatizer para realizar lematización, que convierte las palabras a sus formas base o léxicas.
    - La función stem_and_lemmatize aplica ambos procesos al texto.

- Tokenización: Tokenizamos el texto lematizado y stemmed utilizando word_tokenize de nltk.

- Vectorización utilizando TF-IDF: Utilizamos TfidfVectorizer para convertir el texto a su representación numérica basada en TF-IDF (Term Frequency-Inverse Document Frequency).

- Verificación de Resultados: Mostramos las primeras filas del texto original, texto limpio, texto lematizado y stemmed, texto tokenizado y la matriz TF-IDF para verificar las transformaciones.

Análisis de Sentimientos usando Diccionarios:

En esta etapa, realizamos análisis de sentimientos utilizando dos enfoques basados en diccionarios: Harvard IV-4 con pysentiment2 y Opinion Lexicon con nltk. Estos enfoques permiten determinar el sentimiento de un texto en función de la presencia de palabras positivas y negativas en diccionarios predefinidos.

Estos fueron los resultados:

```
Texto original:
 534435    Well, these capsules are sold by Top Line and ...
52570     No seriously, this is ridiculous, $45 for a ba...
92327     I like strong coffee but was able to use the c...
165543    Started out OK, but after a few weeks my dog d...
201034    The taste of raspberry is pretty much all I ge...
Name: Text, dtype: object

Sentimiento Harvard IV-4:
 534435    0.714286
52570     0.250000
92327     0.500000
165543    0.142857
201034   -0.090909
Name: Harvard_Sentiment, dtype: float64

Sentimiento Opinion Lexicon:
 534435    3
52570     1
92327     5
165543    0
201034   -4
Name: Opinion_Lexicon_Sentiment, dtype: int64
```

Resultados del sexto punto de la practica

- Análisis de Sentimientos usando Harvard IV-4:

- Utilizamos la clase HIV4 de pysentiment2 para tokenizar el texto y obtener una puntuación de sentimiento (Polarity).
- La función analyze_sentiment_harvard aplica este análisis a cada texto en la columna Stemmed_Lemmatized_Text.

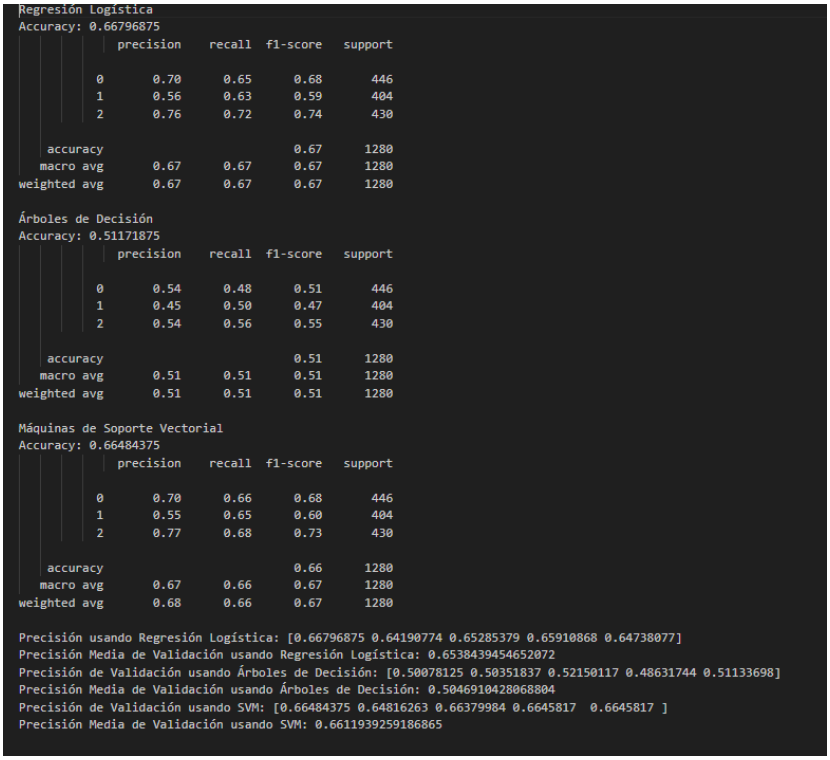
- Análisis de Sentimientos usando Opinion Lexicon:
  - Utilizamos los diccionarios de palabras positivas y negativas de opinion_lexicon de nltk.
  - La función analyze_sentiment_opinion_lexicon tokeniza el texto, cuenta las palabras positivas y negativas, y calcula la diferencia entre ambas para determinar el sentimiento.

- Verificación de Resultados: Mostramos algunas filas del texto original y los resultados de ambos análisis de sentimientos para verificar su precisión.

Análisis de Sentimientos usando Algoritmos de Aprendizaje de Máquina:

En esta etapa, implementamos y evaluamos varios modelos de aprendizaje automático para el análisis de sentimientos, incluyendo Regresión Logística, Árboles de Decisión y Máquinas de Soporte Vectorial (SVM). Además, realizamos una comparación de estos modelos utilizando validación cruzada (k-fold cross-validation).

Estos fueron los resultados:



```
Regresión Logística
Accuracy: 0.66796875
              precision    recall  f1-score   support

           0       0.70      0.65      0.68       446
           1       0.56      0.63      0.59       404
           2       0.76      0.72      0.74       430

    accuracy                           0.67      1280
   macro avg       0.67      0.67      0.67      1280
weighted avg       0.67      0.67      0.67      1280

Árboles de Decisión
Accuracy: 0.51171875
              precision    recall  f1-score   support

           0       0.54      0.48      0.51       446
           1       0.45      0.50      0.47       404
           2       0.54      0.56      0.55       430

    accuracy                           0.51      1280
   macro avg       0.51      0.51      0.51      1280
weighted avg       0.51      0.51      0.51      1280

Máquinas de Soporte Vectorial
Accuracy: 0.66484375
              precision    recall  f1-score   support

           0       0.70      0.66      0.68       446
           1       0.55      0.65      0.60       404
           2       0.77      0.68      0.73       430

    accuracy                           0.66      1280
   macro avg       0.67      0.66      0.67      1280
weighted avg       0.68      0.66      0.67      1280

Precisión usando Regresión Logística: [0.66796875 0.64190774 0.65285379 0.65910868 0.64738077]
Precisión Media de Validación usando Regresión Logística: 0.6538439454652072
Precisión de Validación usando Árboles de Decisión: [0.50078125 0.50351837 0.52150117 0.48631744 0.51133698]
Precisión Media de Validación usando Árboles de Decisión: 0.5046910428068804
Precisión de Validación usando SVM: [0.66484375 0.64816263 0.66379984 0.6645817  0.6645817 ]
Precisión Media de Validación usando SVM: 0.6611939259186865
```

Resultados del septimo punto de la practica

- Conversión de Sentimientos a Números: Convertimos las etiquetas de sentimiento en valores numéricos (Negativo=0, Neutral=1, Positivo=2) usando un mapeo.

- Vectorización usando TF-IDF: Utilizamos TfidfVectorizer para convertir el texto a su representación numérica basada en TF-IDF (Term Frequency-Inverse Document Frequency).

- División en Conjuntos de Entrenamiento y Prueba: Dividimos los datos en conjuntos de entrenamiento y prueba utilizando train_test_split con 0.20 de los datos para prueba.

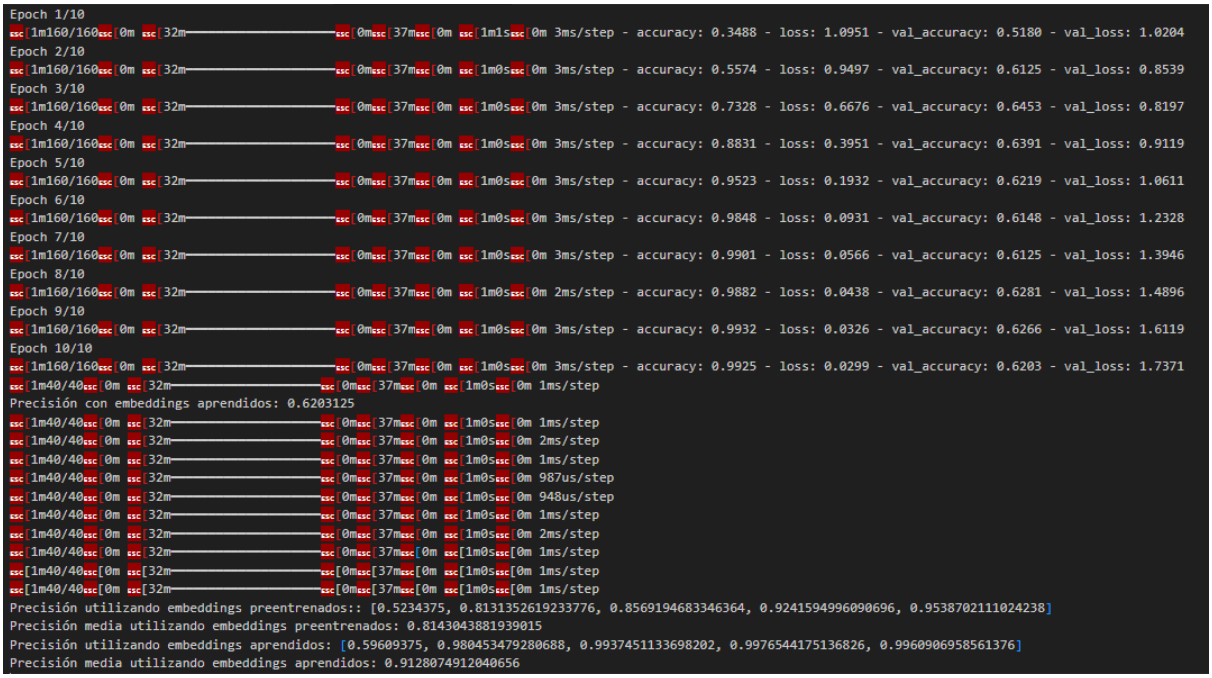
- Entrenamiento y Evaluación de Modelos:
    - Regresión Logística: Entrenamos y evaluamos un modelo de regresión logística.
    - Árboles de Decisión: Entrenamos y evaluamos un modelo de árboles de decisión.
    - SVM: Entrenamos y evaluamos un modelo de máquinas de soporte vectorial.

- Validación Cruzada:
    - Definimos una función para realizar validación cruzada (k-fold cross-validation) y calcular las accuracies de los modelos.
    - Evaluamos los modelos utilizando validación cruzada con k=5 y mostramos las accuracies y sus medias.

Análisis de Sentimientos usando Word Embeddings y Redes Neuronales:

En esta etapa, utilizamos redes neuronales y embeddings preentrenados (GloVe) para realizar el análisis de sentimientos. Implementamos dos modelos: uno que utiliza embeddings preentrenados y otro que aprende los embeddings a partir del conjunto de datos.

Estos fueron los resultados:



Resultados del octavo punto de la practica

- Preparación de Datos:
  - Convertimos las etiquetas de sentimiento a valores numéricos.
  - Tokenizamos y aplicamos padding a las secuencias de texto.
  - Realizamos One-hot encoding de las etiquetas de sentimiento.
  - Dividimos los datos en conjuntos de entrenamiento y prueba.

- Embeddings Preentrenados:
  - Cargamos los embeddings preentrenados de GloVe (50 dimensiones).
  - Creamos una matriz de embeddings utilizando los embeddings de GloVe.
  - Definimos y entrenamos un modelo de red neuronal utilizando los embeddings preentrenados.
  - Evaluamos el rendimiento del modelo en el conjunto de prueba.

- Embeddings Aprendidos:
  - Definimos y entrenamos un modelo de red neuronal que aprende los embeddings a partir del conjunto de datos.
  - Evaluamos el rendimiento del modelo en el conjunto de prueba.

- Validación Cruzada:
  - Definimos una función para realizar validación cruzada (k-fold cross-validation) y calcular las accuracies de los modelos.
  - Evaluamos ambos modelos utilizando validación cruzada con k=5 y mostramos las accuracies y sus medias.

CONCLUSIONES

Análisis de Sentimientos usando Diccionarios

En el análisis de sentimientos utilizando los diccionarios Harvard IV-4 y Opinion Lexicon, se observan diferencias significativas en las puntuaciones de sentimiento obtenidas para las mismas reseñas. Este enfoque es eficiente y rápido, pero su precisión puede verse afectada por la falta de contexto.

Harvard IV-4:

- Las puntuaciones varían entre aproximadamente 0.71 para reseñas positivas a -0.09 para reseñas negativas, reflejando una cierta sensibilidad a la intensidad del sentimiento.
- Sin embargo, no siempre captura matices sutiles o el contexto en el que se utilizan las palabras, que puede generar interpretaciones incorrectas en textos complejos.

Opinion Lexicon:

- Las puntuaciones varían desde 3 (muy positivo) a -4 (muy negativo).
- Este diccionario puede ser demasiado simplista, ya que asigna igual peso a todas las palabras sin considerar su relevancia o contexto específico dentro de la reseña.

Conclusión: Ambos diccionarios ofrecen un análisis rápido y básico del sentimiento, pero su precisión es limitada debido a la falta de consideración del contexto.

Análisis de Sentimientos usando Algoritmos de Aprendizaje de Máquina

Los algoritmos de aprendizaje de máquina, como la regresión logística, árboles de decisión y SVM, muestran una mejora significativa en la precisión del análisis de sentimientos al poder aprender patrones complejos en los datos.

Regresión Logística:

- Alcanzó una precisión de 0.67 en los datos de prueba y una precisión media de validación cruzada de aproximadamente 0.65.
- Es eficaz para identificar relaciones lineales entre las características de los textos y sus sentimientos, siendo relativamente rápido de entrenar y aplicar.

Árboles de Decisión:

- Obtuvo una precisión de 0.51 en los datos de prueba y una precisión media de validación cruzada de aproximadamente 0.56.
- Aunque proporciona interpretaciones claras y visuales de las decisiones, tiende a sobreajustarse a los datos de entrenamiento, lo que puede reducir su rendimiento en datos no vistos.

Máquinas de Soporte Vectorial (SVM):

- Logró una precisión de 0.67 en los datos de prueba y una precisión media de validación cruzada de aproximadamente 0.66.
- Es eficaz para problemas de clasificación donde las clases no son linealmente separables, aunque requiere mayor tiempo de entrenamiento y ajuste de parámetros.

Conclusión: La regresión logística y SVM superan a los árboles de decisión en términos de precisión y capacidad de generalización. Estos modelos son más adecuados para aplicaciones donde se requiere una alta precisión y hay suficiente poder computacional para soportar el entrenamiento y la validación.

Análisis de Sentimientos usando Word Embeddings y Redes Neuronales

El uso de word embeddings y redes neuronales representa una técnica avanzada que captura mejor las relaciones semánticas entre las palabras y mejora significativamente la precisión del análisis de sentimientos con respecto a los modelos anteriormente vistos.

Embeddings Preentrenados (GloVe):

- El modelo con embeddings preentrenados alcanzó una precisión de aproximadamente 0.91 en k-fold cross-validation.
- Estos embeddings proporcionan vectores de palabras que capturan significados semánticos basados en un vasto corpus de texto, mejorando la capacidad del modelo para entender contextos complejos y relaciones entre palabras.

Embeddings Aprendidos:

- El modelo que aprendió embeddings a partir del cuerpo de documentos logró una precisión de aproximadamente 0.62 en k-fold cross-validation.
- Aunque menos precisos que los embeddings preentrenados, estos modelos pueden adaptarse mejor a vocabularios específicos y características estilísticas de los datos de entrenamiento.

Ambos modelos (con embeddings preentrenados y aprendidos) superaron considerablemente a los algoritmos de aprendizaje de máquina en términos de precisión.

Conclusión: Los modelos basados en word embeddings y redes neuronales ofrecen la mayor precisión y son ideales para aplicaciones que requieren un análisis detallado y preciso del sentimiento. Los embeddings preentrenados proporcionaron mejores resultados debido a su capacidad para capturar una amplia gama de significados semánticos a partir de grandes corpus de texto.