****

**Guía de Usuario**

**Análisis de Sentimientos en Ambientes con Comentarios aplicando Procesamiento de**

**Lenguaje Natural NLP**

**Daniel Cortés Pérez**

Universidad Nacional de Colombia

Facultad de Administración, Departamento de Informática y Computación

Manizales, Colombia

2024

**Contenido**

Pág

1. [Introduccion 1](#_Toc112328708)
2. [Guías de Instalación del entorno de desarrollo 2](#_Toc112328709)
3. [Software desarrollado 4](#_Toc112328710)
4. [Fuentes 6](#_Toc112328711)
5. [Repositorio y Acceso 8](#_Toc112328713)
6. [Concluciones 9](#_Toc112328719)

# **1.Introduccion**

Esta guía de usuario proporciona una descripción detallada del sistema desarrollado para el análisis de sentimientos en comentarios utilizando técnicas de Procesamiento de Lenguaje Natural (NLP). Este proyecto, titulado “Análisis de Sentimientos en Ambientes con Comentarios aplicando Procesamiento de Lenguaje Natural NLP”, se enfoca en clasificar comentarios en categorías de sentimientos: positivos, negativos y neutros.

El análisis de sentimientos es fundamental en áreas como marketing, ventas y desarrollo de productos, ya que permite a las organizaciones captar las opiniones y percepciones de sus clientes de manera efectiva. El sistema desarrollado utiliza algoritmos de NLP y clasificación para procesar grandes volúmenes de datos textuales, identificando patrones emocionales y categorizando los comentarios según su polaridad.

El algoritmo implementado, se basa en la Regresión Logística, se emplea principalmente en este caso para analizar comentarios provenientes de reseñas y redes sociales, facilitando la comprensión de las tendencias de opinión pública. Esta información es crucial para la toma de decisiones en ámbitos empresariales y académicos, ofreciendo una visión más profunda sobre las actitudes y preferencias de los usuarios.

El objetivo de esta guía es brindar a los usuarios una orientación clara y detallada sobre cómo instalar el entorno de desarrollo, utilizar el software, explorar sus funcionalidades y comprender los resultados obtenidos. También se incluyen enlaces al repositorio del proyecto para su descarga y modificación, promoviendo la continuidad del trabajo y su adaptación a otros contextos o aplicaciones.

# **2. Guía de Instalación del Entorno de Desarrollo**

**Preparación y manipulación de datos:**

* **Numpy:** Proporciona la base para la computación numérica en Python, ofreciendo estructuras de datos de alta performance como arrays multidimensionales y herramientas para operaciones matemáticas sobre estos.
* **Pandas:** Construye sobre Numpy para ofrecer estructuras de datos más flexibles y fáciles de usar, como Series y DataFrames, diseñadas específicamente para el análisis de datos. Permite cargar, manipular y limpiar datos de diversas fuentes.

**Visualización de datos:**

* **Seaborn:** Una biblioteca de alto nivel basada en Matplotlib, que simplifica la creación de gráficos estadísticos informativos y visualmente atractivos.
* **Matplotlib.pyplot:** Una biblioteca de bajo nivel que ofrece un control más granular sobre la creación de gráficos personalizados.

**Procesamiento de texto:**

* **sklearn.feature\_extraction.text:** Proporciona herramientas para extraer características numéricas de texto, como la frecuencia de palabras o TF-IDF, que son esenciales para muchos modelos de aprendizaje automático basados en texto.
* **CountVectorizer y TfidfVectorizer:** Convierten colecciones de documentos en matrices de recuento de características. CountVectorizer simplemente cuenta la frecuencia de las palabras, mientras que TfidfVectorizer normaliza los recuentos de frecuencia de términos para reflejar la importancia de un término dentro de un documento en relación con el corpus.

**Modelado:**

* **sklearn.linear\_model:** Implementa una variedad de modelos de regresión lineal, incluyendo regresión logística, que es ampliamente utilizada para problemas de clasificación binaria.
* **LogisticRegression:** Un clasificador lineal para problemas de clasificación binaria. Calcula la probabilidad de que una muestra pertenezca a una determinada clase.

**Evaluación de modelos:**

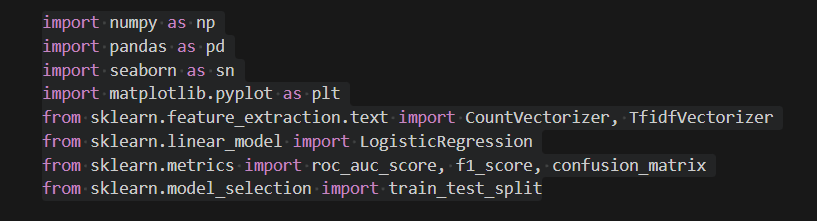
* **sklearn.metrics:** Ofrece una amplia gama de métricas para evaluar el rendimiento de modelos, incluyendo métricas de clasificación (como precisión, recall, F1-score) y métricas de regresión (como error cuadrático medio).
* **roc\_auc\_score, f1\_score, confusion\_matrix:** Métricas específicas para evaluar modelos de clasificación. ROC AUC mide el rendimiento general de un clasificador, F1-score es la media armónica de precisión y recall, y la matriz de confusión proporciona una tabla que muestra las predicciones correctas e incorrectas de un clasificador.

**Preparación de datos para entrenamiento:**

* **sklearn.model\_selection import train\_test\_split:** Divide un conjunto de datos en un conjunto de entrenamiento y un conjunto de prueba, lo que es esencial para evaluar el rendimiento generalizado de un modelo y evitar el sobreajuste.

**Configuración del Entorno de Trabajo:**

* Abrir Jupyter Notebook desde la terminal o consola utilizando el comando “jupyter notebook.”
* Crear un nuevo cuaderno (New > Python 3) para comenzar a trabajar con el código.
* Importar las bibliotecas con comandos simples dentro del cuaderno como:



# **3. Software Desarrollado:**

El código desarrollado utiliza las bibliotecas mencionadas anteriormente: Pandas, Numpy, Seaborn, Matplotlib.pyplot, sklearn.feature\_extraction.text, sklearn.linear\_model, sklearn.metrics, sklearn.model\_selection. El software desarrollado implementa un sistema de análisis de sentimientos utilizando NLTK y sklearn.

**Acceso al Repositorio del Código:**

El código se encuentra disponible en un repositorio público en Git hub en el siguiente link :

## <https://github.com/Dacortesp11/TrabajoDeGradoNLP>

**Instrucciones para Clonar el Repositorio desde GitHub:**

* Abrir la terminal o línea de comandos en el sistema operativo.
* Navegar hasta la carpeta o directorio donde se desea guardar el repositorio clonado usando el comando cd ruta/de/la/carpeta.
* Copiar el enlace del repositorio desde GitHub: <https://github.com/Dacortesp11/TrabajoDeGradoNLP>

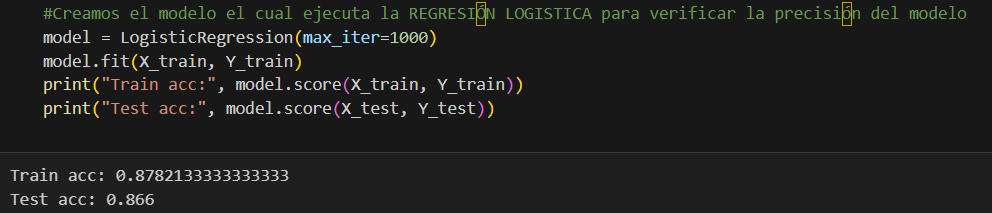
El software desarrollado se centra en la implementación de un sistema de análisis de sentimientos que permite la clasificación automática de comentarios textuales en categorías de sentimientos. Las principales características del software incluyen:

**Preprocesamiento de Datos**

* **Limpieza y Normalización:** Se eliminan caracteres especiales, URLs y menciones, y se normaliza el texto a minúsculas.
* **Tokenización:** Se separa el texto en palabras individuales utilizando NLTK.
* **Lematización:** Se convierte cada palabra a su forma base para reducir la dimensionalidad.

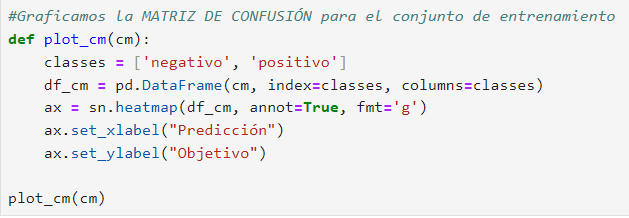
**Entrenamiento de Modelo de Clasificación**

* **Regresión Logística:** Se entrena un modelo de regresión logística para la predicción de sentimientos basándose en la frecuencia de términos en el texto.

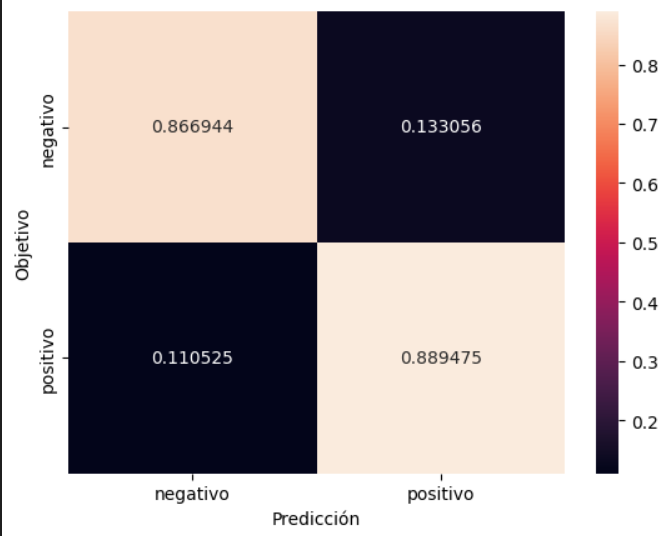


# **Evaluación y Visualización**

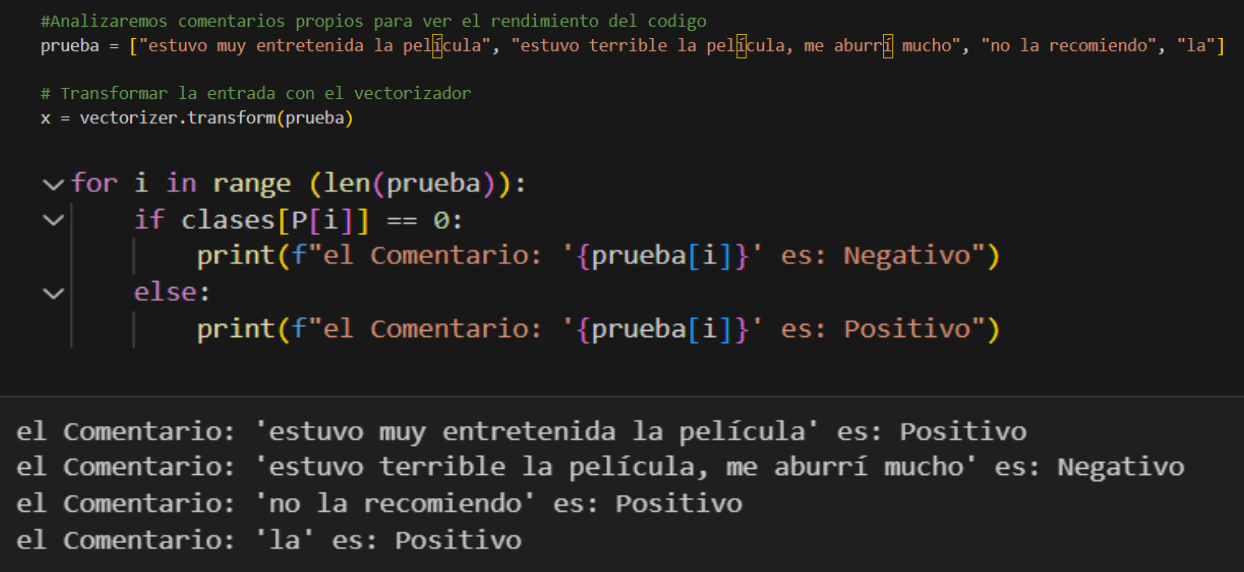
* **Matrices de Confusión:** Para evaluar el rendimiento de cada modelo en términos de precisión, sensibilidad y especificidad.
* **Gráficos ROC y AUC:** Para evaluar la capacidad predictiva de los modelos.
* **Visualización de Sentimientos:** Gráficos que muestran la distribución de los sentimientos en el conjunto de datos.



# **Matriz de confusión – gráfica:**



# **Breve resumen de funcionamiento del algoritmo:**



# **4. Preprocesamiento y Manipulación de Datos**

# **Recolección de Datos**

**Se utilizaron dos fuentes principales para la recolección de datos:**

* **Reseñas de Películas:** Datos extraídos de una página web que contiene opiniones de usuarios sobre diversas películas.
* **Comentarios de Twitter:** Recopilación de tweets utilizando la API de Twitter para obtener opiniones sobre temas específicos.

**Proceso de Limpieza y Transformación**

* **Eliminación de Ruido:** Se remueven caracteres no textuales, como enlaces y menciones a usuarios.
* **Normalización:** Conversión de texto a minúsculas y eliminación de stopwords.
* **Vectorización:** Conversión del texto a vectores numéricos utilizando la técnica TF-IDF (Term Frequency-Inverse Document Frequency).

# **5. Evaluación y Visualización de Resultados**

# **Métricas Utilizadas**

# **Precisión:** Proporción de verdaderos positivos sobre el total de predicciones positivas.

# **Recall:** Proporción de verdaderos positivos sobre el total de elementos verdaderamente positivos.

# **F1 Score:** Media armónica de precisión y recall, útil para evaluar modelos con clases desbalanceadas.

# **Visualización de Resultados**

# Se generaron gráficos que muestran la efectividad de los modelos, incluyendo:

# **Curvas ROC y AUC:** Para medir la capacidad de discriminación del modelo entre diferentes clases.

# **Matrices de Confusión:** Para visualizar los errores de clasificación cometidos por cada modelo.

# **Distribución de Sentimientos:** Gráficos que muestran la proporción de comentarios positivos, negativos y neutros en el conjunto de datos.

# **Repositorio y Acceso**

# El software desarrollado se encuentra alojado en un repositorio público en GitHub para facilitar el acceso y la colaboración. Se puedes encontrar el repositorio en el siguiente enlace:

* <https://github.com/Dacortesp11/TrabajoDeGradoNLP>

**Nombre del repositorio en GitHub:** TrabajoDeGradoNLP

**Acceso al Repositorio:** Para acceder al repositorio y utilizar el software desarrollado, seguir estos pasos:

* Visitar el enlace del repositorio:   
  **TrabajoDeGradoNLP** en GitHub.
* Explorar los archivos y la estructura del repositorio.
* Clonar o Descargar el repositorio completo como un archivo ZIP utilizando el botón "Code".

# **7. Conclusiones y Recomendaciones**

**Conclusiones del Proyecto**

**Resultados Satisfactorios:** Se logró desarrollar un modelo eficiente para el análisis de sentimientos en comentarios textuales, alcanzando una precisión de hasta el 88% en la clasificación de sentimientos positivos.

**Limitaciones Identificadas:** Se observó una confusión significativa en la clasificación de comentarios neutros, lo que sugiere la necesidad de mejorar la representación de esta categoría en el conjunto de datos.

# **Recomendaciones para Futuros Trabajos**

# **Ampliar el Conjunto de Datos:** Incluir más fuentes de datos y comentarios en diferentescontextos para mejorar la robustez del modelo.

# **Optimización de Modelos:** Explorar el uso de redes neuronales y técnicas de deep learning para mejorar la precisión en la clasificación de sentimientos.

# **Implementación en Tiempo Real:** Adaptar el sistema para su uso en tiempo real, permitiendo la clasificación de comentarios a medida que se generan en plataformas en línea.

# **8. Referencias**

**Bibliotecas y Recursos Utilizados:**

**-Pandas**: McKinney, W. (2010). Data Structures for Statistical Computing in Python. Proceedings of the 9th Python in Science Conference.

Sitio web: Pandas

**-Matplotlib**: Hunter, J.D. (2007). Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering.

Sitio web: Matplotlib

**-NumPy**: Oliphant, T.E. (2006). A guide to NumPy. Trelgol Publishing.

**Sitio web:** NumPy

**-SK-learn**: Pedregosa, F. et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research.

Sitio web: Scikit-Learn

**-Seaborn**: Michael Waskom. (2021). Seaborn: Statistical Data Visualization. Journal of Open-Source Software.

Sitio web: Seaborn

Liu, B. (2012). Sentiment Analysis and Opinion Mining. Morgan & Claypool Publishers.

Pang, B., & Lee, L. (2008). Opinion Mining and Sentiment Analysis. Foundations and Trends in Information Retrieval.

Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. Ain Shams Engineering Journal, 5(4), 1093-1113.

Hutto, C. J., & Gilbert, E. (2014). VADER: A parsimonious rule-based model for sentiment analysis of social media text. Proceedings of the International AAAI Conference on Web and Social Media.

Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python. O'Reilly Media.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. Advances in Neural Information Processing Systems, 26.