TÍTULO DEL PROYECTO: Diagnóstico de Experiencia del Cliente en E-commerce mediante Deep Learning
FECHA: 22 de Noviembre de 2025

1. RESUMEN EJECUTIVO

El objetivo de este proyecto ha sido desarrollar una solución de Inteligencia Artificial capaz de diagnosticar las causas raíz de la fricción en el comercio electrónico. A diferencia de los sistemas tradicionales que solo analizan una puntuación general (estrellas), esta propuesta implementa un **Análisis de Sentimiento Basado en Aspectos (ABSA)** utilizando Deep Learning.

La solución técnica consiste en una red neuronal secuencial **Bi-LSTM** diseñada para clasificar simultáneamente el sentimiento en tres dimensiones críticas del negocio: **Producto, Logística y Servicio**.

Los resultados obtenidos superaron los objetivos establecidos, alcanzando una exactitud promedio superior al 95% en el conjunto de validación y demostrando, mediante pruebas cualitativas, la capacidad de desacoplar sentimientos mixtos.

2. DATOS: OBTENCIÓN Y DISPONIBILIDAD

Para garantizar la viabilidad computacional, la reproducibilidad de los experimentos y la persistencia de los archivos, se estableció la siguiente estrategia de gestión de datos:

Fuente de Datos y Repositorio Se utilizó como base el "Synthetic E-commerce Product Reviews Dataset" .

Fuente Original: Disponible públicamente en la plataforma Kaggle .

Repositorio del Proyecto (GitHub): Para facilitar la corrección y el despliegue, tanto el dataset original como el dataset procesado (`ecommerce_dataset_100k_BALANCED.csv`) se han incluido en el repositorio de GitHub adjunto a esta entrega. Esto permite clonar el proyecto y ejecutar los notebooks sin depender de descargas externas temporales.

Disponibilidad y Acceso en los Notebooks El flujo de datos se diseñó para ser flexible en la nube (Google Colab):

Ingesta: El proceso admite la carga manual del archivo `ecommerce_product_reviews_dataset.csv` o la lectura directa desde la carpeta del repositorio clonado.

Muestreo: Para optimizar el rendimiento, se reduce el dataset a una muestra aleatoria de **100,000 registros** .

Generación Sintética (Augmentation): El dataset final utilizado para el entrenamiento se genera y guarda dentro de la misma sesión. Este archivo final (`_BALANCED.csv`) es el que alimenta al modelo y se encuentra respaldado en el GitHub para futuras inferencias.

3. DESCRIPCIÓN DE LA ESTRUCTURA DE LOS NOTEBOOKS

El desarrollo del proyecto se ha modularizado en **dos notebooks principales** para mantener el código limpio y organizado:

Notebook 1: Análisis Exploratorio de Datos (EDA) Este cuaderno se centra exclusivamente en la comprensión de los datos brutos antes de cualquier modelado.

Objetivo: Diagnosticar la calidad del dataset original y justificar las decisiones de diseño.

Contenido Clave:

- Inspección de tipos de datos y valores nulos.
- Visualización de la distribución de calificaciones (detectando el desbalance de clases).
- Análisis de frecuencias de palabras (WordClouds) para entender el vocabulario del e-commerce.
- Este análisis fue la base para decidir la implementación de la técnica de *Data Augmentation*.

Notebook 2: Implementación del Modelo (Preprocesamiento y Deep Learning) Este es el cuaderno principal de ejecución (End-to-End), donde ocurren las **Fases 2 y 3** de forma secuencial e integrada:

Fase de Ingeniería y Preprocesamiento:

- Ejecución del algoritmo de *Data Augmentation* para balancear el dataset.
- Limpieza de texto y normalización.
- Tokenización y Padding (transformación de texto a secuencias numéricas).

Fase de Modelado y Evaluación:

- Construcción de la arquitectura Bi-LSTM Multi-Salida.
- Entrenamiento del modelo con el dataset procesado en la fase anterior.
- Evaluación final con métricas de negocio y pruebas de inferencia manual.

Esta estructura permite separar el análisis estadístico (Notebook 1) de la tubería de producción del modelo (Notebook 2), facilitando la revisión y el mantenimiento del código.

4. DESCRIPCIÓN DE LA SOLUCIÓN

Para abordar el problema de clasificación multi-salida, se diseñó una arquitectura de **Red Neuronal Recurrente (RNN)** utilizando la API Funcional de Keras. Los componentes clave son:

1. Capa de Embedding:

Transforma cada palabra en un vector denso de 64 dimensiones, permitiendo al modelo capturar relaciones semánticas básicas entre los términos.

2. Bi-Directional LSTM (Bi-LSTM):

Se seleccionó esta arquitectura 1 porque procesa el texto en dos direcciones (de izquierda a derecha y viceversa). Esto es fundamental para entender el contexto completo de una oración, por ejemplo, para distinguir entre "el envío no fue rápido" y "el envío fue rápido".

3. Cabezales de Salida Múltiples (Multi-Head):

A diferencia de un modelo tradicional, esta red cuenta con tres capas de salida independientes 2:

- **Salida 1 (Y1):** Sentimiento del Producto.
- **Salida 2 (Y2):** Sentimiento de la Logística.
- **Salida 3 (Y3):** Sentimiento del Servicio.

Cada salida utiliza una función de activación **Softmax** para clasificar el resultado en tres categorías: Negativo, Neutral o Positivo.

5. DESCRIPCIÓN DE LAS ITERACIONES REALIZADAS

El proyecto requirió iteraciones estratégicas para superar las limitaciones de los datos sintéticos originales:

Iteración 1: Enfoque Heurístico (Fallido) Inicialmente, se intentó etiquetar los aspectos de Logística y Servicio buscando palabras clave en el dataset original. El resultado fue insatisfactorio debido a que el dataset sintético apenas contenía menciones negativas sobre envíos o soporte, resultando en clases vacías que impedían el aprendizaje del modelo.

Iteración 2: Estrategia de Data Augmentation (Exitosa) Se pivotó hacia una estrategia de enriquecimiento de datos. Se desarrolló un algoritmo que inyecta aleatoriamente frases realistas (ej. *"Shipping took forever"*, *"Support was rude"*) en el 30% de las reseñas. Esto permitió balancear las clases y proporcionar al modelo ejemplos claros de fricción en Logística y Servicio, haciendo viable el entrenamiento.

Iteración 3: Optimización de Compilación Se ajustó el código de compilación del modelo, pasando de diccionarios de nombres a listas ordenadas, para evitar errores de compatibilidad en Keras y asegurar que las métricas de precisión se calcularan correctamente para cada una de las tres salidas.

6. DESCRIPCIÓN DE LOS RESULTADOS

Métricas Cuantitativas El modelo fue evaluado en un conjunto de validación independiente (20% de los datos). Se alcanzó una exactitud (Accuracy) promedio superior al **95%**, superando holgadamente el objetivo del 70% establecido en la propuesta inicial.

- Exactitud Producto: ~87%
- Exactitud Logística: ~99%
- Exactitud Servicio: ~99%

(Nota: Los valores excepcionalmente altos en Logística y Servicio se deben a la consistencia de los patrones introducidos durante la fase de Data Augmentation).

Resultados Cualitativos Se realizaron pruebas manuales con frases nuevas para verificar la utilidad del modelo en el negocio. El sistema demostró éxito en casos complejos, como la reseña: *"The product quality is amazing but shipping was very late"*. En este caso, el modelo clasificó correctamente:

- **Producto:** Positivo
- **Logística:** Negativo

Esto confirma que la solución es capaz de identificar fricciones específicas ("Tasa de Identificación de Fricción") sin confundirlas con la opinión general del producto, cumpliendo así con el objetivo de negocio principal