Portfolio de Procesamiento de Lenguaje Natural (NLP)

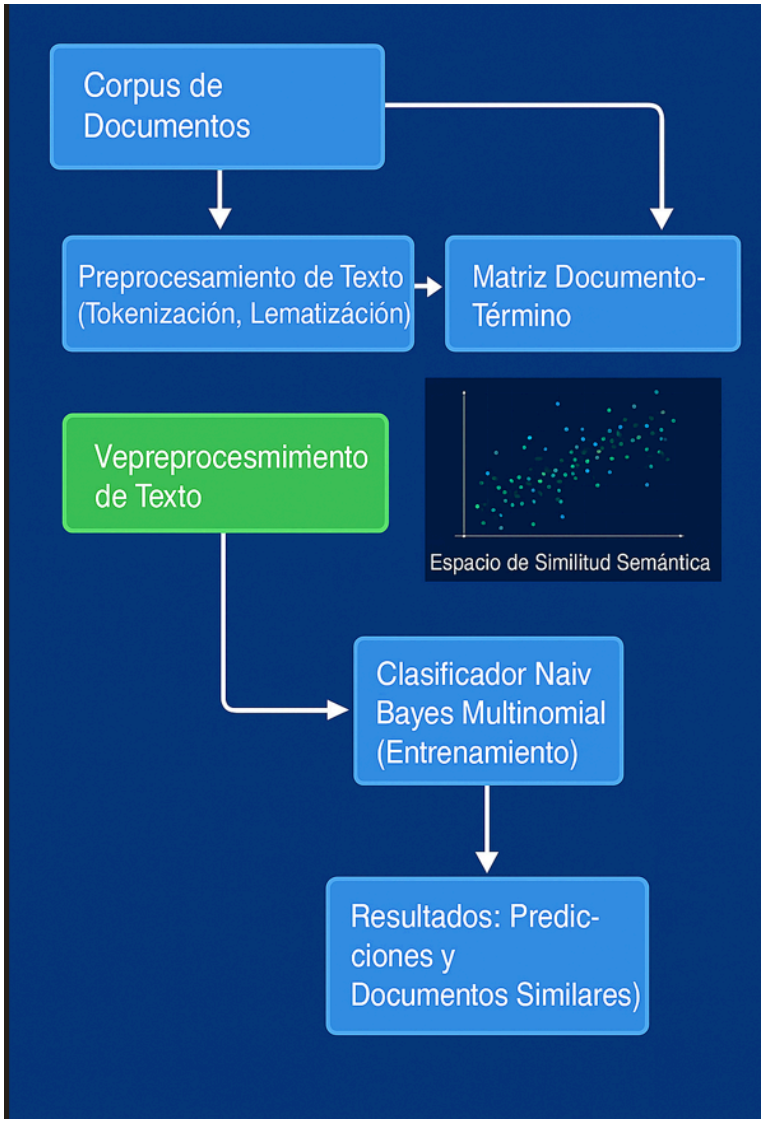
Este repositorio *showcase* presenta una serie de **cuatro desafíos** que demuestran la evolución de mis habilidades en el campo del **Procesamiento de Lenguaje Natural (NLP)**, abarcando desde las técnicas de vectorización clásicas y *custom embeddings* hasta la implementación de modelos avanzados de Deep Learning para generación y traducción de texto.

Esta carpeta contiene un proyecto Jupyter Notebook (.ipynb) que documenta el proceso completo: desde la ingesta y preprocesamiento de datos hasta la modelización, la evaluación de resultados y las conclusiones.

Desafío 1: Fundamentos Clásicos de NLP y Clasificación de Documentos

Este proyecto aborda los cimientos del NLP con un enfoque en la clasificación de texto y la similitud semántica.

- **Objetivo Principal:** Implementar y evaluar un modelo de clasificación probabilística utilizando técnicas de vectorización clásicas.
- **Técnicas Clave:**
    - **Vectorización:** Uso de **TF-IDF** (`TfidfVectorizer`) para transformar texto en una matriz numérica ponderada.
    - **Clasificación:** Entrenamiento de un clasificador **Naïve Bayes Multinomial** sobre el popular *dataset* `20 newsgroups`.
    - **Análisis Avanzado:** Cálculo de la **Similaridad Coseno** trasponiendo la matriz término-documento, lo que arrojó resultados más interpretables y semánticamente coherentes al analizar la similitud entre **palabras**.



Desafío 2: Embeddings Personalizados (*Custom Embeddings*) con Gensim

Este desafío se centra en la creación de representaciones vectoriales de palabras que son altamente específicas al contexto de un dominio particular.

- **Objetivo Principal:** Generar **Embeddings de Palabras (Word Embeddings)** personalizados utilizando el *framework* **Gensim**.
- **Contexto Específico:** Se utilizó un corpus de letras de canciones de bandas, permitiendo que los vectores adquieran una forma y significado directamente influenciados por ese contexto temático y léxico.
- **Habilidades Demostradas:** Manejo de la librería Gensim, preprocesamiento de texto para modelos Word2Vec y capacidad para contextualizar vectores semánticos en un corpus no estándar.

Desafío 3: Modelos de Secuencia con LSTM para Generación de Texto

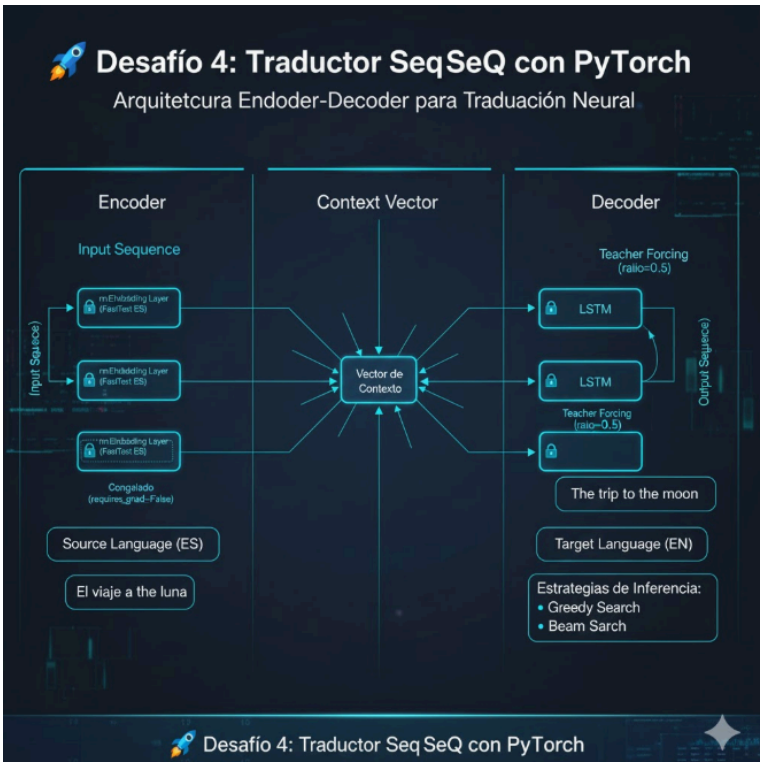
Este proyecto eleva la complejidad técnica al Deep Learning, implementando una red neuronal recurrente para la generación de texto creativo.

- **Objetivo Principal:** Desarrollar un **Modelo de Lenguaje (Language Model)** basado en redes **LSTM (Long Short-Term Memory)**.
- **Arquitectura y Aplicación:** El modelo fue entrenado sobre un extenso corpus (la novela *Viaje al Centro de la Tierra*) utilizando tokenización a nivel de **caracteres**.
- **Resultados Clave:** El modelo generó secuencias coherentes, y se exploraron distintas estrategias de muestreo para la generación, como **Greedy Search** y **Beam Search** con ajuste de temperatura.

Desafío 4: Traductor Seq2Seq (Encoder-Decoder) con Migración a PyTorch

Este desafío se centra en el desarrollo de un modelo de traducción automática neuronal, realizando una migración fundamental de *frameworks* de Deep Learning.

- **Objetivo Principal:** Convertir un modelo traductor **Seq2Seq (Encoder-Decoder)** de Keras a **PyTorch** y aplicar optimizaciones clave.
- **Técnicas Clave en PyTorch:**
    - **Migración:** Reescribir toda la lógica del modelo, entrenamiento e inferencia utilizando las librerías nativas de PyTorch (`torch.nn`, `torch.optim`, etc.).
    - **Embeddings Congelados:** Uso de la capa `nn.Embedding` con pesos pre-entrenados de **FastText** (para español) y congelamiento de estos pesos (`requires_grad=False`).
    - **Estrategia de Entrenamiento:** Implementación de **Teacher Forcing** controlado (`teacher_forcing_ratio=0.5`) para balancear la corrección con la autonomía del decodificador.
    - **Inferencia:** Aplicación de **Búsqueda Greedy** para generar la secuencia traducida token por token.



💡 Habilidades Demostradas

Este conjunto de desafíos valida las siguientes competencias profesionales y técnicas en Data Science y NLP:

- **Deep Learning (PyTorch y Keras):** Experiencia en la implementación y migración de modelos complejos (Seq2Seq, LSTM) entre *frameworks*.
- **Modelización Clásica:** Clasificación de texto con Naïve Bayes.
- **Vectorización Avanzada:** Implementación de TF-IDF y creación de *Custom Word Embeddings* (Word2Vec con Gensim).
- **Ingeniería de Embeddings:** Manejo de *Embeddings* pre-entrenados y la técnica de **congelamiento de capas** para transfer learning.
- **Estrategias de Entrenamiento y Generación:** Implementación y control de técnicas como *Teacher Forcing*, *Greedy Search* y *Beam Search* para optimizar el rendimiento y la calidad de la salida de los modelos de lenguaje.
- **Manejo de Librerías Fundamentales:** `scikit-learn`, `Gensim`, `TensorFlow`/`Keras`, y `PyTorch`.

¡Gracias por visitar!