REPORTE EJECUTIVO

Presentado por: GRUPO 1

CINDY JOHANNA ZAPATA ROMERO

HECTOR GEOVANY BELLO SANTAMARÍA

MARIO GUERRA GUALY

LEANDRO REYES JORDÁN

Docente:

CARLOS ISAAC ZAINEA MAYA

UNIDAD DE ESTUDIO:

GERENCIA DE PROYECTOS PARA CIENCIA DE DATOS

MAESTRÍA - GRUPO 1 - M1V - VIRTUAL - 2025

FACULTAD DE INGENIERÍA

BOGOTÁ, 01 DE MARZO DE 2025

UNIVERSIDAD EAN

INTRODUCCIÓN

El propósito de este informe es lograr evidenciar el aprendizaje del proceso de gestión de proyectos en ciencia de datos. Partiendo del análisis del caso de estudio, definimos un sistema de gestión híbrido, usando metodología agiles como SCRUM y la planeación del estilo CRIPS-PM como las metodologías para establecer los lineamientos y estrategias necesarios para alcanzar el objetivo del proyecto.

Así pues, definimos 3 scripts principales como parte de la metodología ágil, que partieron de las historias de usuario, maduradas con la definición de épicas y luego detalladas en actividades y entregables. Con este trabajo inicial, se crean los recursos necesarios para entender las necesidades del proyecto, el análisis de riesgos, la definición de roles y también como se manejará la gobernanza de los datos.

Plasmando y organizando esta información, lo que resta es la ejecución y gestión del proyecto. Las personas con el conocimiento adecuado habrán realizado el análisis y preparación de los datos, creación de modelos de clasificación y el control de los entregables que permitan expresar los resultados esperados por los stakeholders en indicadores y gráficos sencillos.

METODOLOGÍA

Ciclo de Vida del Proyecto - CRISP-DM

La gestión del proyecto se realizó siguiendo CRISP-DM (Cross Industry Standard Process for Data Mining), que consta de las siguientes fases:

Comprensión del Negocio: Se definieron objetivos y requerimientos del proyecto basados en las necesidades del stakeholder.

- **Comprensión de los Datos**: Se exploraron y analizaron las características del dataset seleccionado.
- **Preparación de los Datos:** Se limpiaron los datos, aplicando transformaciones necesarias para garantizar su calidad.
- **Modelado:** Se implementaron modelos de clasificación RNN y LSTM para evaluar su desempeño.
- **Evaluación**: Se midieron los resultados obtenidos para determinar la efectividad de cada modelo.
- **Despliegue**: Se documentaron los hallazgos y se generaron recomendaciones para futuras iteraciones.

Gestión del Proyecto - Scrum

Para la gestión del proyecto, se utilizó Scrum, estructurando el trabajo en tres sprints con las siguientes características:

- **Sprint 1:** Definición de requerimientos, recopilación de datos, primera exploración y preprocesamiento inicial.
- **Sprint 2:** Desarrollo de los modelos de clasificación, ajuste de hiperparámetros y evaluación preliminar.

- **Sprint 3:** Refinamiento del modelo, optimización del preprocesamiento y generación de reportes.

Se definieron seis historias de usuario, organizadas en dos épicas, y se estimaron 52 puntos de historia en total. Se usaron herramientas como Trello para la gestión de tareas y GitHub para la documentación y el control de versiones.

Análisis de Datos y Resultados

Se detalla el flujo de trabajo para clasificar noticias, abarcando desde la exploración del dataset hasta el diseño, entrenamiento y evaluación de modelos basados en RNN y LSTM.

2. Exploración y Análisis del Dataset

- ✓ Origen y Estructura: Se importa un dataset en Excel a un DataFrame de Pandas con un tamaño de 14.396 filas y 6 columnas (enlace, título, contenido y etiqueta).
- ✓ Análisis Descriptivo: Conteo de artículos por categoría para evaluar el balance de clases, revisando los valores nulos e inconsistencias, garantizando la calidad de los datos.

3. Preprocesamiento del Texto

- ✓ Limpieza y Normalización: Conversión a minúsculas, eliminación de caracteres especiales y números para lograr la Tokenización o secuencias numéricas de textos.
- ✓ Estandarización: Aplicación de padding para uniformar la longitud de las secuencias.
- ✓ Filtrado de Etiquetas: Selección de las categorías más representativas, reduciendo el ruido y optimizando el entrenamiento.

4. Arquitectura de los Modelos: Se evalúan dos enfoques de modelado:

4.1. Modelado en RNN

- Arquitectura:
  - ✓ Capa de Embedding para representación vectorial.
  - ✓ Capa SimpleRNN (64 neuronas) para procesar secuencias, con limitaciones en capturar dependencias largas por desvanecimiento del gradiente.
  - ✓ Capa Dense con activación Softmax para clasificación.

Resultados y Desafíos: Accuracy de entrenamiento que varía del 66% al 93% en pocas épocas, con caídas en evaluación, evidencia la limitación de la arquitectura.

4.2. Modelado en LSTM

- Definición de la Arquitectura:
  - ✓ Capa de Embedding: Convierte cada palabra en un vector denso de 528 dimensiones, capturando relaciones semánticas.
  - ✓ Capa Bidireccional LSTM: Procesa la secuencia en ambas direcciones, mejorando la comprensión del contexto.
  - ✓ Capa Dense con Softmax: Clasifica asignando una probabilidad a cada categoría.

Parámetros y Resultados: Uso del optimizador Adam y función de pérdida categorical_crossentropy, adicional la configuración bidireccional mejora la captación de dependencias con nuevas y anteriores entradas en la configuración de los vectores densos. Manteniendo un crecimiento del accuracy hasta el 97% en entrenamiento y del 86% en la validación.

5. Entrenamiento y Evaluación

- ✓ Métricas: Se reportan accuracies en entrenamiento y validación para ambos modelos.
- ✓ Visualización: Gráficos de accuracy para el entrenamiento y la validación permiten identificar fenómenos como sobreajuste y desvanecimiento del gradiente.

7. Conclusiones

- ✓ Comparativa: El modelo LSTM muestra mayor estabilidad y capacidad para capturar contextos complejos, mientras que el RNN, a pesar de alcanzar altos accuracies en entrenamiento, se ve limitado por el desvanecimiento del gradiente.
- ✓ Recomendaciones: Ajustar técnicas de regularización, optimizar hiperparámetros y explorar arquitecturas híbridas para mejorar la generalización, especialmente del modelo RNN.

CONCLUSIONES Y RECOMENDACIONES

El análisis de modelos de clasificación de noticias demostró que LSTM supera a RNN en la captura de contextos largos, logrando 97% de precisión en entrenamiento y 86% en validación, mientras que RNN sufrió desvanecimiento del gradiente, limitando su capacidad de aprendizaje. A pesar del buen desempeño de LSTM, se evidenció sobreajuste, lo que indica la necesidad de mejorar la regularización y optimización de hiperparámetros.

El preprocesamiento de datos y la reducción de ruido fueron fundamentales para la calidad del modelo, aunque la dimensionalidad de los embeddings y la longitud de secuencias requieren ajustes adicionales. Además, la integración de Scrum y CRISP-DM facilitó la estructuración del desarrollo, aunque la validación temprana del modelo debe fortalecerse en futuros proyectos.

Recomendaciones

a. Optimización del Modelo:

- Implementar Dropout (0.3 - 0.5) y Batch Normalization para mitigar el sobreajuste en LSTM.
- Ajustar la tasa de aprendizaje con Scheduler Learning Rate para estabilizar la convergencia del modelo.

b. Mejoras en el Preprocesamiento de Datos:

- Reducir la dimensionalidad de los embeddings con PCA o Autoencoders para optimizar la representación semántica.
- Ajustar la longitud de secuencias mediante análisis de percentiles para evitar padding innecesario.

c. Evaluación y Comparación de Modelos:

- Aplicar matrices de confusión y SHAP/LIME para interpretar mejor las predicciones.
- Optimizar hiperparámetros con Optuna o Hyperopt para mejorar la eficiencia del modelo.
- Explorar arquitecturas híbridas combinando LSTM con mecanismos de atención o Transformers ligeros para mejorar la generalización.