**UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS**

**(DECANA DE AMÉRICA)**

**FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

**ESCUELA PROFESIONAL DE INGENIERÍA DE SOFTWARE**



**Documentación para Ev Continua - Data**

**CLASIFICACIÓN DE PUBLICACIONES**

**CLASIFICACIÓN DE PUBLICACIONES**

1. **Objetivo del issue/ tarea / task:**

El objetivo de la tarea es desarrollar un modelo de procesamiento de lenguaje natural (NLP) capaz de clasificar las publicaciones de la red social en diferentes categorías o temas, como política, educación, deporte, economía y otros. Este modelo permitirá etiquetar automáticamente las publicaciones según su contenido temático, facilitando así la organización y búsqueda de contenido en la plataforma.

1. **Requerimientos de task:**

* R\_P01: Recopilar y preparar un conjunto de datos etiquetado que contenga un conjunto representativo de publicaciones con sus respectivas categorías o temas asignados.
* R\_P02: Realizar un preprocesamiento de texto adecuado, que incluya la tokenización, eliminación de stop words y otros pasos necesarios para limpiar y estructurar los datos de manera que sean adecuados para el modelado de NLP.
* R\_M03: Elegir y configurar un modelo de clasificación de texto, como una red neuronal, un modelo de Bag of Words (BoW) o un modelo basado en Transformer, y ajustar los hiperparámetros de acuerdo con la naturaleza de la tarea.
* R\_M04: Dividir el conjunto de datos en conjuntos de entrenamiento y prueba para evaluar el rendimiento del modelo.
* R\_P05: Entrenar el modelo de clasificación utilizando el conjunto de entrenamiento y ajustarlo hasta alcanzar un rendimiento satisfactorio en términos de precisión en la clasificación de textos por categorías o temas.
* R\_P06: Evaluar el rendimiento del modelo utilizando métricas apropiadas, como precisión, recall, F1-score y matriz de confusión, en el conjunto de prueba.
* R\_P07: Documentar y presentar los resultados obtenidos, incluyendo una interpretación de las métricas y una descripción de las categorías o temas a las que se asignaron las publicaciones.
* R\_P08: Integrar el modelo en la plataforma de la red social para automatizar la clasificación de publicaciones por categorías o temas.
* R\_P09: Realizar pruebas exhaustivas del modelo en un entorno de producción para garantizar su rendimiento y precisión en condiciones reales.
* R\_M10: Proporcionar documentación técnica y un manual de uso para que otros miembros del equipo puedan comprender y mantener el modelo en el futuro.

1. **Modelo elegido y parametrización**

**Nombre del Modelo:** Como grupo, hemos decidido utilizar BERT (Bidirectional Encoder Representations from Transformers) como nuestro modelo principal.

Parametrización del Modelo:

Para esta tarea de clasificación de textos por categorías o temas, estamos utilizando BERT como nuestro modelo de referencia. BERT es un modelo de lenguaje pre-entrenado que ha demostrado un alto rendimiento en una variedad de tareas de procesamiento de lenguaje natural (NLP). A continuación, describimos cómo estamos parametrizando este modelo para nuestro proyecto:

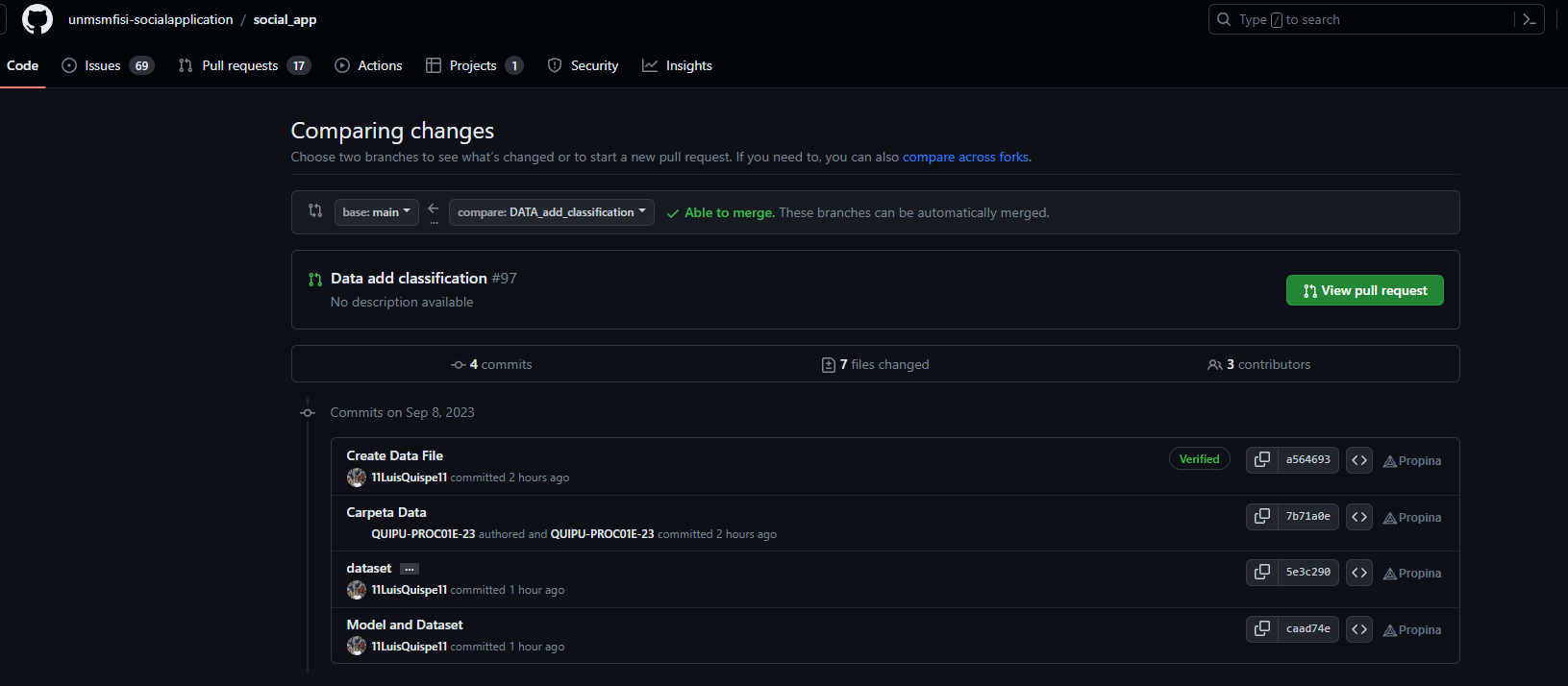
**Ajuste de la Capa de Salida**: Hemos personalizado la capa de salida del modelo BERT para que coincida con el número de categorías o temas que deseamos clasificar como grupo. Utilizamos una capa completamente conectada (fully connected) para adaptar el modelo a nuestra tarea específica.

**Entrenamiento de la Capa de Salida**: Una vez ajustada la capa de salida, estamos entrenando el modelo en nuestro conjunto de datos etiquetado. Durante el entrenamiento, ajustamos los pesos de la capa de salida para que el modelo pueda aprender a clasificar correctamente las publicaciones en las categorías o temas deseados.

**Fine-Tuning:** Como grupo, estamos considerando la opción de fine-tuning (ajuste fino) en la que entrenamos el modelo BERT completo en nuestra tarea específica.

**Hiper Parámetros de Entrenamiento:** Hemos ajustado los hiper parámetros de entrenamiento, como la tasa de aprendizaje, el tamaño del lote (batch size) y el número de épocas, de acuerdo con la complejidad de nuestra tarea y los recursos disponibles.

1. **Subida del scrip (Pull request , commit)**



1. **Resultados y discusion**

Métricas de Evaluación:

F1-Score: 0.9215

Recall: 0.8980

Interpretación de Resultados:

F1-Score: El valor del F1-Score es alto (0.9215), lo que indica que el modelo tiene un buen equilibrio entre precisión y recall. Un F1-Score cercano a 1 generalmente indica un buen rendimiento en la clasificación.

Recall: El recall es del 89.80%, lo que significa que el modelo es capaz de recuperar aproximadamente el 89.8% de los casos positivos reales. Esto indica una sólida capacidad del modelo para detectar casos positivos.

Conclusión del Modelo y su Aplicación:

Basado en los resultados obtenidos, podemos concluir que el modelo tiene un rendimiento sólido en la tarea de clasificación. El alto F1-Score y el buen recall sugieren que el modelo es efectivo en la detección de casos positivos, lo que lo hace adecuado para aplicaciones en las que la identificación precisa de casos positivos es crítica. Sin embargo, es importante considerar el contexto específico de la aplicación y los requisitos para determinar si el rendimiento del modelo satisface las necesidades del problema en cuestión.

**Nota: Lo qué vendria adjuntando un responsable del modelo seria , el script de Python , requirements.txt (este ultimo tengo entendido son las librerias y su version , por el momento ignorarlo)**