Descripcino usuario y aplicación:

El usuario principal de la aplicación sería una organización como el UNFPA, junto con otras entidades públicas que buscan recopilar, analizar y procesar opiniones de ciudadanos en torno a temas como la salud, la educación y la igualdad de género, relacionados con los ODS 3, 4 y 5.

La aplicación ayudaría a estos organismos a identificar patrones y prioridades sociales, brindando datos importantes que podrían ayudar en la toma de decisiones políticas para la población, de tal forma que se solucionen problemáticas presentadas en el país.

De esta manera, la importancia que representa la aplicación se ve reflejada en la automatización del análisis de grandes volúmenes de datos con las opiniones de los ciudadanos. Datos los cuales, suelen ser extensos y complejos de categorizar de manera sencilla, por los dialectos manejados y el contexto de cada uno de los ciudadanos que participan.

Como se menciona anteriormente, la aplicación ayudaría para enviar cada opinión, contexto y problema al área acorde, mostrando los porcentajes asociados del texto con cada una de las categorías impuestas.

Así mismo, la pagina ayuda a tener una categorización más certera de cada una de las opiniones por medio del modelo, lo que podría incurrir en disminución de gastos y tiempo para las entidades al manejar menos personal humano en la división de dichos contextos.

En ese orden de ideas, la pagina se conecta de manera directa con el proceso de evaluación y monitoreo de las ODS, debido a que su uso le permite a las entidades identificar de manera rápida, sencilla y efectiva las áreas mayormente involucradas en cada una de las opiniones dadas por los ciudadanos. Esto ayudando a encontrar problemáticas dentro de la población y de esta forma, buscar soluciones optimas que ayuden a mitigarlas.

Estrategias para el reentrenamiento

En primer lugar, optamos por tomar tres posibles estrategias que fueran las mejores para el reentrenamiento del modelo, entre estas elegimos el incremental, el reactivo y por último, uno basado en umbral de rendimiento.

Primero, el reentrenamiento reactivo se refiere a un enfoque en el que el modelo se vuelve a entrenar como respuesta directa a algún tipo de evento o condición específica en el entorno de los datos o el modelo mismo. Como por ejemplo cambios en los datos, teniendo un evento conocido como cambio de concepto, que quiere decir cambios en la distribución o patrones de los datos. Tambien cuando se muestra nueva información disponible para el modelo.

Aunque, la desventaja que encontramos en la estrategia para ser usada por nosotros fue que no sigue un plan predefinido de cuándo o cómo debe reentrenarse, sino que se vuelve a entrenar como una medida correctiva ante una condición que requiere acción inmediata.

En segundo lugar, el reentrenamiento basado en umbral de rendimiento se basa en la definición de métricas clave de rendimiento como el F1-score y la monitorización continua del modelo en producción. Si el rendimiento del modelo cae por debajo de un umbral predefinido en alguna de estas métricas, se dispara el proceso de reentrenamiento. Aunque para nuestro caso, es una opción mucho más compleja de poder implementar para una api sencilla usada en este caso, por tal motivo, no fue nuestra preferida tampoco.

Finalmente, teniendo en cuenta que para la empresa, la recopilación constante de nuevas opiniones. De tal manera, que los ciudadanos muestren, comenten y expresen sus pensamientos, estos nuevos datos obtenidos pueden afectar a gran medida los resultados del entrenamiento como en las percepciones y prioridades de la población. El reentrenamiento incremental permite ajustar el modelo para tener en cuenta estas nuevas opiniones sin necesidad de entrenarlo nuevamente desde cero, manteniendo la eficiencia del proceso. Así mismo, el lenguaje y las expresiones de las personas pueden cambiar a través del tiempo, como adquirir nuevas palabras o temas. Al mantener un modelo de reentrenamiento que le permita adaptarse a dichos cambios verbales sin desechar lo anteriormente aprendido, permite mantener la efectividad del modelo; Y por último, como se espera que la aplicación funcione de manera continua, el reentrenamiento incremental permite que el modelo se actualice de manera fluida sin necesidad de detener el servicio.