MBIT School

BUSINESS DATA INTELLIGENCE

TÍTULO: Predicción de ODS's (Radar ODS)

AUTORES: Guillermo Pueyo & Josep Velasco

FECHA: 15/12/2021

Índice general

Índice de figuras

Índice de tablas

Repositorio Github

https://github.com/GuillePueyo/Radar-ODS-Proyecto-MBIT.git

1. Resumen Ejecutivo

Los Objetivos de Desarrollo Sostenible (ODS) constituyen un llamamiento universal a la acción para poner fin a la pobreza, proteger el planeta y mejorar las vidas y las perspectivas de las personas en todo el mundo.

En 2015, todos los Estados Miembros de las Naciones Unidas aprobaron 17 Objetivos como parte de la Agenda 2030 para el Desarrollo Sostenible, en la cual se establece un plan para alcanzar los Objetivos en 15 años.

Con el fin de poder hacer un seguimiento real del progreso de los ODS (así se llamarán a los Objetivos de Desarrollo Sostenible de ahora en adelante) y poder "medir" el estado en el que nos encontramos, se ha creado un Proyecto que se encargará de este proceso. Para ello, se ha colaborado con la plataforma eAgora. Se trata de una aplicación móvil para ser el sitio de unión, conexión y encuentro digital entre la administración y la ciudadanía, entidades y comercios.

La metodología del Proyecto consiste en el uso de algoritmos de Procesamiento de Lenguaje Natural (Machine Learning) para predecir ODS's haciendo uso de la información ofrecida en la aplicación móvil.

El "core" del Proyecto es el procesamiento de lenguaje natural (que llamaremos NLP de ahora en adelante) y se enfoca como un problema de Machine Learning Supervisado. Esto quiere decir que se necesitan unos datos iniciales para poder crear y entrenar nuestro algoritmo.

El dataset de datos utilizado ha sido creado y etiquetado artificialmente. El número de etiquetas son 17 (una por cada ODS)

Como conclusión al Proyecto, se puede confirmar que el predictor de ODS funciona bien cuando el comentario no es demasiado largo, pero funciona a una baja probabilidad cuando el comentario es más largo dado que el dataset utilizado es muy pequeño.

2. Introducción

eAgora es una "startup" creada en Tarragona en el año 2019. Está diseñada para ser el lugar de unión, conexión y encuentro digital entre la administración y la ciudadanía, entidades, ONG, comercios y empresas.

A través de eAgora, las administraciones facilitan a los ciudadanos poder:

Consultar:

Gestiona: facilitar varios trámites como presentar una instancia, descargar documentos oficiales, solicitar espacios públicos, etc.

Vive: apoyar en el día a día con todo el que hay cerca de mi

Informa: poder consultar noticias, agenda, alertas y web y apps municipales

Participar:

Canales: varias temáticas enlazadas con los ODS. Nos podemos subscribir a una categoría (por ejemplo: Cambio Climático), seguir un canal sugerido (Reciclaje) y finalmente participar en el canal (hacer propuestas concretas). Además de poder saber "Qué sucede en mi ciudad".

Módulos extra y escalabilidad: nuevas soluciones basadas en necesidades del ayuntamiento o por petición de la ciudadanía (previa aceptación de la administración)

Actuar:

Reportar incidencia: Medidas correctoras e imprevistas.
Enviar información: Alimentar el banco de recursos a través de experiencias conocidas

El Proyecto que se va a dar a cabo, consiste en poder vincular de manera automática cada uno de los canales que se crean en la aplicación de móvil con los diferentes ODS. Al finalizar este Proyecto se podrá ofrecer un valor añadido a la plataforma ya que ahora mismo ninguno de los posibles competidores está ofreciendo este servicio a los clientes.

2.1 Motivación del Proyecto

Con el fin de apoyar a la medición de los ODS a nivel nacional este Proyecto es un buen candidato para ello. Habiendo acordado la colaboración con la plataforma eAgora, se disponen de los canales necesarios para conectar con los ayuntamientos, ciudadanos, comercios y entidades.

A partir de esta App, se puede estudiar los comentarios de actividades o iniciativas que se publiquen en la App a nivel nacional y etiquetarlos en el ODS correcto, si procede.

2.2 Hipótesis iniciales / Problemas a resolver

La *hipótesis 1* fue usar los datos de prueba existentes dentro de la plataforma para poder etiquetarlos.

Los datos eran demasiado pequeños para ser utilizados para entrenar un modelo de Machine Learning. Literalmente había 160 comentarios por lo que se realizaron reuniones con la empresa para valorar un aumento de la información por parte de ellos.

La *hipótesis 2* fue la de utilizar los comentarios de la plataforma cuando ya había muchos más. Había unos 1500+ comentarios para poder ser etiquetados y utilizados para entrenar nuestro modelo.

La *hipótesis 3* fue la de crear manualmente un dataset artificial de calidad y usarlo para entrenar el posible modelo. Esto es un trabajo más costoso, sobre todo de tiempo ya que hay que escribir uno por uno y asignarle su etiqueta correcta.

En cuanto al problema que se quería resolver, se trataba de una predicción de texto multi-etiqueta, es decir, un texto podría tener más de una etiqueta (predicción)

2.3 Previsiones de desarrollo

La plataforma eAgora está creciendo muy rápido y se estima que la cantidad de datos que tendrá que procesar será muy grande.

Una de las principales previsiones de desarrollo es usar la misma metodología de Machine Learning para predecir comentarios, pero llevado a un entorno Cloud propuesta hecha en el punto de *Arquitectura de la solución*.

2.4 Relación con el Máster de Big Data

La relación con el Máster en Big Data es muy completa. Se han usado herramientas vistas en el máster además de usar el lenguaje de programación más utilizado durante todo el Máster lo cual hace que este haya sido tremendamente útil.

La relación será mayor cuando se pueda completar el punto 2.3 ya que se tendrá que implementar herramientas como podría ser AWS, Google Cloud o Azure (ambientes cloud)

2.5 Técnicas de referencia

Las técnicas utilizadas están relacionadas con el NLP. En este caso en particular, se necesitaba de un algoritmo que no sólo predijera un solo ODS sino también varios ya que hay acciones que pueden relacionarse a más de un ODS. Por ejemplo, la siguiente frase;

"*Estamos creando una iniciativa para donar alimentos para los más necesitados"*

En este caso, esta frase puede hacer referencia al ODS1 (Fin de la pobreza) y ODS2 (Fin del hambre).

Para poder afrontar este tipo de problema, se tuvo que recurrir a la *clasificación de textos multi – etiqueta.(Multi-label en inglés)*

No se debe confundir multi-clase con multi-etiqueta. Para ello existen varias imágenes que pueden ilustrar esto y así entenderlo mejor;

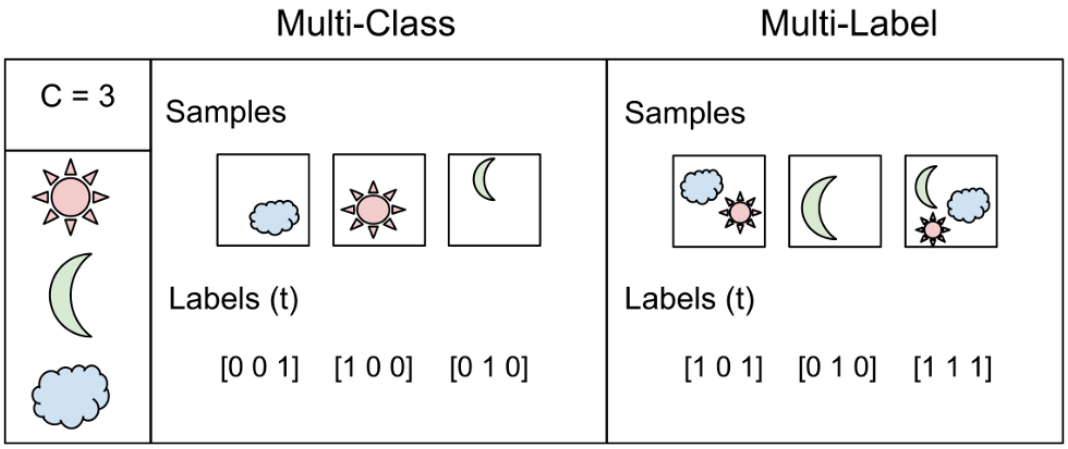

Fig. 1. Esquema de diferenciación entre Multi-Class y Multi-Label

Como se puede observar claramente, el problema que se plantea es Multi-Label (derecha) ya que, para el mismo comentario, se puede tener más de una etiqueta.

La diferencia entre el problema multi-clase y multi-etiqueta es que en los problemas de multi-clase las clases son mutuamente excluyentes, mientras que para los problemas de multi-etiqueta cada etiqueta representa una tarea de clasificación diferente, pero las tareas están relacionadas de alguna manera.

En el proyecto el problema vendría de la siguiente manera. Se puede observar que para varios comentarios, existen varias etiquetas;

	ODS1	ODS2	ODS3	ODS4	ODS5	ODS6	ODS7	ODS8	ODS9	ODS10	ODS11	ODS12	ODS13	ODS14	ODS15	ODS16	ODS17
Comentario 1	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0
Comentario 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
...																	
...																	
Comentario N	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Fig. 2. Tabla de comentarios con varios ODS identificados.

3. Estado del Arte

Actualmente no existe ninguna empresa que haga el estudio de ODS's tan directo como se pretende hacer con el actual proyecto propuesto por la plataforma de eAgor.

Tampoco existen trabajos orientados a la predicción de ODS usando técnicas de Machine Learning. Se podría decir que el Proyecto de estudio roza la novedad ya que se va a extraer predicciones de ODS directamente de las personas, ayuntamientos o comercios.

Existen numerosas empresas que se dedican a ayudar a otras empresas a alcanzar ciertos retos relacionados con los ODS. Se pueden describir a continuación;

.- D - Good People; se trata de una herramienta SaaS para digitalizar el impacto de ESG (E – Environment, S – Social y G – Governance) de los empleados y mejorar la reputación sostenible de las empresas.

Esta empresa solo ayuda a mejorar cuatro ODS's;

ODS 4 – Educación de calidad
ODS9 – Industria e innovación
ODS10 – Reducción de las desigualdades
ODS17 – Alianzas para lograr los objetivos

.- #COMPANIES4SDG; en este caso no se puede considerarlo empresa, se trata de una asociación sin ánimo de lucro que tiene varios objetivos;

1.- Promover los Objetivos de Desarrollo Sostenible entre las empresas y sus empleados.

2.- Sensibilizar a la población sobre los **retos globales del mundo** y la forma en que pueden participar.

3.-Fomentar hábitos sostenibles entre los empleados y trasmitirles cómo **contribuir a los retos** globales de los ODS.

4.- Ofrecer a las empresas la oportunidad de sumarse al principal movimiento social global de la mano de expertos, de otras empresas y con el apoyo de IMPACT 2030.

5.- Alinear y fomentar un **Voluntariado Corporativo** alineado a los ODS, a través de una campaña global.

También existen medios de comunicación para dar a conocer los ODS y hacer seguimiento de ellos. Un ejemplo es el caso de *ÁGORA - Inteligencia colectiva para la sostenibilidad* que surge para trabajar por una responsabilidad Social incluyente, por un mundo sostenible donde la sensibilización, la educación para el cambio y el consumo sostenible y responsable sean el pilar del nuevo paradigma, donde podamos crear convicción en base a la razón y la fuerza de los argumentos y no en base al poder económico.

4. Propuesta de viabilidad

Los datos que se han utilizado para este proyecto son los comentarios que escriben los usuarios de la aplicación en cada canal que se crea. Éstos se guardan en una tabla en el servidor que eAgora tiene contratado y en S3 de Amazon Web Service (AWS).

El acceso a ellos es mediante una conexión a su servidor donde sólo se tiene permisos de lectura y que está permitido descargar la tabla para trabajar en local.

Una vez se disponen de la tabla hay que hacer un primer recorrido por cada uno de los registros ya que existe la posibilidad de que estén en dos idiomas,

castellano o catalán. Este hecho ha provocado el primer problema a la hora de etiquetar los textos ya que el modelo tiene que distinguir entre los dos idiomas.

Por otro lado, otro aspecto para tener en cuenta es la calidad de los textos. Los signos de puntuación, faltas, y otro tipo de signos pueden afectar a la hora de obtener un resultado satisfactorio. Por ello, se debe realizar una limpieza antes de aplicarle el modelo. Para ello se han utilizado las herramientas disponibles en las diferentes librerías existentes en el lenguaje de Python.

Por otro lado, existe la posibilidad de que un mismo comentario y/o mismo canal pueda pertenecer a dos o más ODS, hecho que dificulta más a la hora de hacer el modelo ya que se tuvo que escoger realizar el trabajo mediante clasificación *multi-etiqueta*

El volumen de datos viene en función de la actividad que cada administración y/o usuario haga de la propia aplicación. Actualmente la plataforma creada por eAgora se encuentra implementada en municipios pequeños, pero ya existen preacuerdos con ciudades más importantes. Este hecho provocará que la base datos vaya creciendo de manera exponencial y dentro del mismo proyecto se plantea una solución para su escalabilidad.

Estos dos últimos aspectos son los que tendrán un mayor peso específico a la hora de realizar los estudios económicos ya que para soportar el aumento de datos y tener una mayor escalabilidad, se propone utilizar estructuras Cloud, pudiendo aumentar el actual coste de mantenimiento de la infraestructura.

5. Arquitectura de la solución

A nivel de arquitectura no existe un gran trabajo ya que todo está bien sujeto. Se ha tenido que entrenar un modelo para poder predecir los comentarios provenientes de una aplicación.

Esta aplicación inserta estos comentarios mediante una API a una tabla de SQL.

En la siguiente imagen se ilustra;

Figura 3. Arquitectura de la solución

5.1 Propuesta de solución Cloud

A continuación, se va a describir la arquitectura que se va a proponer a eAgora con la visión de crecimiento que se prevé que tendrá dicha plataforma. Aunque la mayor parte del proyecto se ha focalizado en la búsqueda del mejor algoritmo que prediga el o los ODS más adecuados para cada tipo de comentario, se ha querido realizar esta propuesta de arquitectura del dato.

5.1.1-Ciclo del dato

El dato tal y como se ha mencionado en puntos anteriores se crea cuando un usuario escribe en alguno de los canales de la aplicación móvil. Éste actualmente se carga en una tabla y se guardan en los servidores de eAgora.

El paso que se propone es realizar la carga de todos los datos que se generan en la aplicación al servicio de S3 de Amazon Web Service (AWS). Aunque actualmente las tablas no tienen un peso elevado, se tienen unas perspectivas de crecimiento considerables sobre todo cuando se ejecuten los contratos de grandes ciudades y los que puedan venir en un futuro. Esto provocará que las tablas aumenten y evitaremos así posibles problemas de escalabilidad.

Dentro del servicio S3 se creará un *bucket* y en él se crearán dos carpetas, una que se llamará “input/” donde se irían cargando la tabla de datos que se generarán cada día en la plataforma. Y otra carpeta que se llamará “output/” donde se cargará la tabla con los ODS salientes del algoritmo de cálculo y que se utilizará para los Dashboards.

Por otro lado, se crearía el servicio *lambda* para realizar la tarea de pretratado de los textos y la aplicación del modelo para encontrar los ODS.

Para el pretratado de los textos, se trabajará con las stop words, con los signos de puntuación, mayúsculas, etc. Para ello se utilizarán y se importarán las librerías *re* y *string*. Y, por otra parte, se seguirá con otro bloque de código para poder identificar el idioma. Para ello, se va a utilizar la librería *langdetect* y se va a importar *detect* y *DetectorFactoryse*.

Esta *lambda* se activaría con un evento mediante el triggers de S3. Este evento se va a relacionar con el acto de incorporar un archivo en el *bucket* y dentro de la carpeta “input/” creado anteriormente en S3. Y para que se conecte al servicio se va a utilizar la API de AWS, la librería *boto3*.

Se creará otra lambda la cual se activaría cada día a las 24h mediante CloudWatch. En ella se aplicaría el código del algoritmo para encontrar los ODS de los textos del último archivo entrado en la carpeta “input/”. Y después añadiría el resultado al archivo ya creado con los ODS y que se encontraría guardado en la carpeta “output/”.

Y por último, para poder mostrar los datos, utilizaremos la aplicación PowerBI. Con ella se va a conectar el archivo guardado en el *bucket* de S3, en la carpeta de “output/”, dándole a éste los permisos públicos para que se pueda tener acceso desde el exterior.

5.1.2.-Esquema del Ciclo del dato

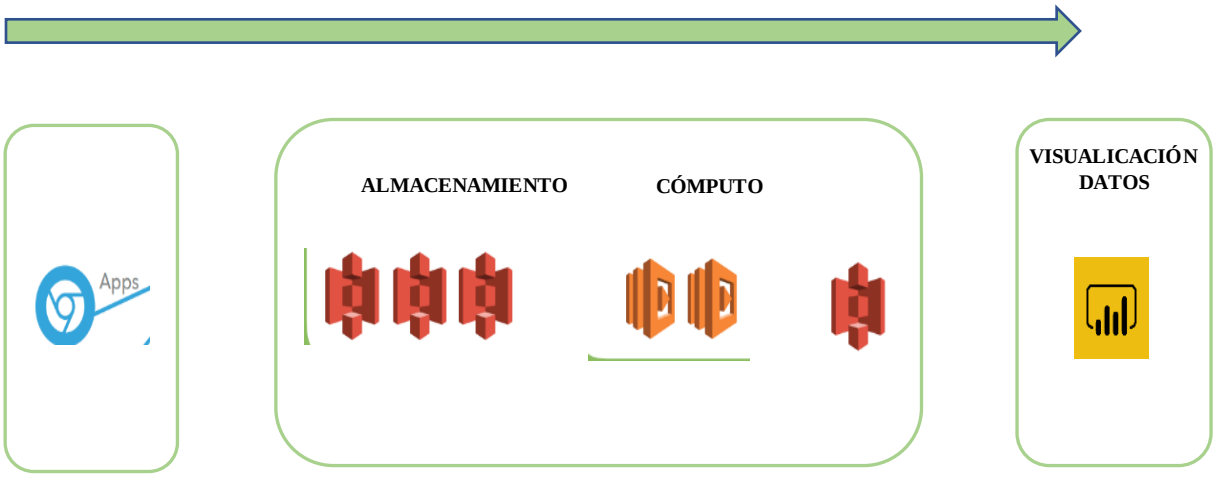

Figura 4. Esquema de la arquitectura Cloud en Amazon Web Service (AWS)

5.1.3.-Modelo de precios de los servicios de AWS

El modelo de precios de estos servicios se definirá en la siguiente tabla con las hipótesis correspondientes

SERVICIO	PRECIO	HIPÒTESIS	PRECIO FINAL
S3 (Almacenamiento)	0,0023$/GB	Por la previsión de crecimiento se supondrá un valor de almacenamiento de 100 GB	0,23$
S3 (Transerencia a PowerBI)	0,09$/GB	Suposición de que la tabla guardada con los ODS pese 100 GB	9 $
Lambda	0,2$ por 1M de peticiones 512MB 0,000000834$ por 100ms	1000 peticiones/día 1000 ms tiempo ejecución del código 512 MG de memoria	free
TOTAL			**9,23 $**

Tabla 1. Propuesta económica de la implantación de la arquitectura Cloud en AWS.

6. Plan de trabajo

El plan de trabajo ha sido realizado por dos personas. Cada uno tenía una tarea asignada con ciertas fechas de entrega para poder así aligerar el proceso.

Hay que decir que las reuniones de proyecto entre el equipo han sido todas online ya que cada uno vive en una provincia distinta.

Durante los meses de verano (Julio y Agosto) el proyecto ha estado prácticamente parado, retomando las actividades en Septiembre.

En la tabla de debajo se pueden ver las distintas etapas del plan de trabajo. En un siguiente paso, se van a detallar cada una de las etapas;

PLANNING DE TRABAJO

Tareas	Actividades	Descripción	Marzo	Abril	Mayo	Junio	Julio	Agosto	Sept	Oct	Nov
Kick off	Primera Reunión con eAgora	*Reunión para conocer al equipo y fijar el alcancel del proyecto*									
Meetings	Segunda reunión con eAgora	*Reunión con el equipo de tecnología para revisar su arquitectura y sistemas que utlizan*									
	Tercera reunión con eAgora	*Reunión con el equipo de tecnología para darnos acceso al servidor y a sus tablas*									
Dataset	Etiquetado del dataset existente	*Etiquetado a mano del dataset existente en su BBDD para usarlo en el entreno del modelo*									
	Dataset no válido	*Después de invertir días en el etiquetado, decidimos no usarlo para entrenar el modelo ya que es poco consistente y los resultados que arroja son muy malos*									
	Creación de dataset artificial	*Decidimos crear un dataset artificial a mano, creando frases por nosotros y también usando titulares de noticias relacionadas con cada ODS*									
Modelado	Problema del idioma	*Se barajaron varias posibilidades de afrontarlo y llevó algo de tiempo la decisión*									
	Estudio del modelo	*Se trata de un modelo multi-etiqueta, menos común que los vistos en clase por lo que tuvimos que invertir tiempo en entenderlo más a fondo*									
Maquillar presentación TFM	Maquillar presentación TFM	*Crear una presentación para la defensa del TFM además de varios ejemplos de visualización*									
Memoria TFM	Memoria TFM	*Redactar Memoria TFM*									

Figura 5. Temporización del plan de trabajo

6.1.- Meetings:

Las reuniones fueron de una hora aproximada de duración donde estuvimos discutiendo los siguientes apartados:

- Conocer al equipo colaborador y discutir las posibles direcciones del proyecto ya que al principio el objetivo era la creación de una “alarma” para hacerles saber a los ayuntamientos que alguna de las partes que componen una calle o plaza (banco, semáforo, papeleras, etc) estaba en mal estado o roto. (**cambio de rumbo del proyecto**)

- Presentación de la parte de tecnología y backend de la aplicación además de dashboards donde hacían sus visualizaciones de la información.

- Instalación de la VPN que se va a utilizar para acceder a sus tablas (solo lectura) y confirmar que podíamos acceder a ellas mediante Python y My SQL Workbench

Para llevar a cabo estas reuniones, nos llevó bastante tiempo (unos 2 meses) para completarlas y poder tener algo sólido donde empezar.

6.2.- Dataset:

Para poder obtener el dataset inicial, se tuvo que esperar aproximadamente un mes y medio ya que los primeros registros que se disponían eran de 163 comentarios almacenados en una tabla de SQL. Esta tabla de SQL se actualiza por cada registro que entra en la aplicación en uno de los módulos.

- Dataset no válido (**callejón sin salida**);

1.- Una vez que se estuvo en espera de aproximadamente un mes y medio para poder tener más información en la tabla, los resultados llegaron hasta 1553 registros.

2.- Estos registros tenían que ser etiquetados con los posibles 17 ODS’s.

3.- Se comenzó con el etiquetado hasta llegar al registro 500.

4.- Después de ese etiquetado, se realizó una pequeña prueba con un modelo multi-etiqueta pero solo predecía el ODS 3 (Salud y bienestar)

5.- Al estar tan desbalanceado el dataset, se decidió crear uno artificial de alta calidad.

6.3.- Creación del dataset artificial:

Después de muchas horas de trabajo de etiquetado en el dataset original, se decidió crear uno artificial de alta calidad para comenzar y poder en un futuro hacerlo más grande para incrementar la calidad.

Se realizó un dataset manualmente de 1500 registros bien etiquetados. Se basó en los siguientes puntos:

- Escribir frases típicas donde lanzábamos iniciativas de colaboración en ciertas actividades.

- Duplicar la misma frase mediante el cambio de algunas palabras por sinónimos.

- Búsqueda de noticias enfocadas al ODS que estuviéramos etiquetando en ese momento.

6.4.- Problema del idioma:

La empresa colaboradora es de origen catalán y por ende, es una aplicación móvil donde el catalán estará en cualquier sitio.

En este caso, se tuvo que hacer la traducción de todos los comentarios que se hicieron previamente en castellano. Una vez hecha esa traducción, se tuvo que entrenar la propuesta de modelo en un idioma distinto por lo que finalmente se tienen dos modelos de predicción; castellano y catalán.

En este caso surgió la necesidad de analizar cómo proceder cuando un comentario estaba en castellano y otro en catalán. Mediante el uso de Python con su librería de identificación de idiomas se llegó a una buena solución.

7. Resultados

Puesto que el dataset del proyecto es muy pequeño, los rendimientos de los modelos no han sido muy altos, pero sí suficientes para hacer predicciones con cierto criterio.

También hay que remarcar que las posibles clases a predecir son 17, demasiado grande para un dataset de tan solo 1453 líneas.

7.1.- Datasets generados y su potencial

1. Dataset proporcionado por la empresa

La empresa colaboradora proporcionó un dataset que no estaba en buenas condiciones para ser tratado. Se trataba de un dataset de 1500 líneas (comentarios) que en muchos casos no tenían nada que ver con la predicción de ODSs y por otro lado estaba muy desbalanceado, sobre todo con el ODS3 (Saludo y bienestar). Aún así, el primer paso fue empezar a etiquetarlo y analizar después si tenía potencial o no.

Este primer dataset estaba además en un solo idioma (catalán) con frases aleatorias en castellano.

El rendimiento de los modelos con este dataset será mostrado en el siguiente apartado.

2. Dataset artificial

Con el paso del tiempo vimos que el dataset proporcionado por la empresa no iba a ser de calidad y nuestro modelo no iba a ser bueno. Ya que el dataset iba a ser pequeño, por lo menos que fuera de alta calidad.

Por tanto, se por la creación de un dataset manualmente y totalmente artificial. Se fue creando cada comentario una a uno y etiquetado. Los comentarios eran tanto inventados como titulares de periódicos digitales.

Aproximadamente se pudo crear 1453 comentarios (85 – 90 comentarios por ODS). Con la creación de este dataset se miró de asegurar de que es perfecto para un modelo de predicción en términos de contenido, pero no de tamaño.

7.2 *Modelos aprendidos y su rendimiento*

A la hora de crear el modelo, se ha tenido que analizar varios para ver cuál era el más adecuado. En nuestro caso, al tener un dataset tan pequeño, no va a tener tantas diferencias entre un modelo u otro.

- **ML_KNN – Vecinos más cercanos**

Este método es una adaptación del tradicional KNN de clasificación de una sola etiqueta. En Python existen estos módulos que hacen las aproximaciones al problema de multi-etiqueta.

- **Relevancia Binaria con Naive Bayes**

En este caso, se entrena un conjunto de clasificadores binarios de etiqueta única, uno para cada clase. Cada clasificador predice la pertenencia o la no pertenencia de una clase. La unión de todas las clases que se predijeron se toma como salida de etiquetas múltiples.

Esto se puede ver mejor con un ejemplo más visual. Si se imagina que tenemos el dataset de datos etiquetados;

	ODS1	ODS2	ODS3	ODS4	ODS5	ODS6	ODS7	ODS8	ODS9	ODS10	ODS11	ODS12	ODS13	ODS14	ODS15	ODS16	ODS17
Comentario 1	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0
Comentario 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
...																	
...																	
Comentario N	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Fig. 6. Tabla de comentarios con varios ODS identificados.

En relevancia binaria, la idea consiste en separarlo en 17 casos de clasificación única, como sigue;

X	ODS1	X	ODS2	X	ODS3	....	X	ODS17
Comentario 1	0	*Comentario 1*	0	*Comentario 1*	0	....	*Comentario 1*	0
Comentario 2	0	*Comentario 2*	0	*Comentario 2*	0		*Comentario 2*	1
...		...		...		....	...	
...		...		...			...	
Comentario N	1	*Comentario N*	1	*Comentario N*	0		*Comentario N*	0

Fig. 7. Tabla de separación de los comentarios con los ODS identificados.

Esta técnica está implementada en Python y es la que se ha usado para crear el modelo.

- **Label Powerset con Regresión Logística**

Este método busca las coincidencias en términos de etiqueta para después hacer la transformación que veremos a continuación. De momento se marcarán en color amarillo y azul los comentarios con las mismas etiquetas.

X	ODS1	ODS2	ODS3	ODS4	ODS5	ODS6	ODS7	ODS8	ODS9	ODS10	ODS11	ODS12	ODS13	ODS14	ODS15	ODS16	ODS17
Comentario 1	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0
Comentario 2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Comentario 3	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0
Comentario 4	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0
...																	
...																	
Comentario N	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Comentario N+1	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0

Fig. 8. Tabla de comentarios con varios ODS identificados.

Por lo tanto, se va a transformar este problema en un solo problema de varias clases, como se muestra a continuación.

X	Y
Comentario 1	1
Comentario 2	2
Comentario 3	3
Comentario 4	4
...	
...	
Comentario N	2
Comentario N+1	1

Fig. 9. Tabla de comentarios transformados con varios ODS identificados.

Por lo tanto, label powerset ha asignado una clase única a cada combinación de etiquetas posible que está presente en el conjunto de entrenamiento.

Una vez explicado los 3 métodos que se ha usado para crear nuestros modelos y evaluar el rendimiento, pasamos a mostrar las métricas obtenidas tanto usando un dataset en castellano como el mismo dataset en catalán.

Se ha empezado por el *accuracy_score.* Esta es una métrica muy sencilla que representa el porcentaje total de valores correctamente clasificados, tanto positivos como negativos.

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + TN + FN)}$$

Fig. 10 Ecuación de la Accuracy.

Aparte de esta métrica, debemos evaluar otras más específicas;

- **Micro –Average Precision**; Es la suma de los verdaderos positivos para todas las clases divididas por todas las predicciones positivas y se representa con la siguiente fórmula;

$$PrecisionMicroAvg = \frac{(TP_1 + TP_2 + \ldots + TP_n)}{(TP_1 + TP_2 + \ldots + TP_n + FP_1 + FP_2 + \ldots + FP_n)}$$

Fig. 11. Ecuación de la Precisión Micro-Average.

- **Macro –Average Precision**; se puede definir como la media aritmética de todas las puntuaciones de precisión de diferentes clases. Se representa con la siguiente fórmula;

$$PrecisionMacroAvg = \frac{(Prec_1 + Prec_2 + \ldots + Prec_n)}{n}$$

Fig. 12. Ecuación de la Precisión Macro-Average.

- **Micro – Average Recall;** La suma de los verdaderos positivos para todas las clases dividida por los positivos reales (y no los positivos previstos). Se representa de la siguiente manera;

$$RecallMicroAvg = \frac{(TP_1 + TP_2 + \ldots + TP_n)}{(TP_1 + TP_2 + \ldots + TP_n + FN_1 + FN_2 + \ldots + FN_n)}$$

Fig. 13. Ecuación Recall Micro-Average.

- **Macro – Average Recall;** La media aritmética de todas las puntuaciones de recall de diferentes clases. Matemáticamente la representamos así;

$$RecallMacroAvg = \frac{(Recall_1 + Recall_2 + \ldots + Recall_n)}{n}$$

Fig. 14. Ecuación Recall Macro-Average.

Una vez detalladas todas las métricas que vamos a valorar, dejaremos en una frase el significado de precision, recall y F1;

- **Precisión** nos da la calidad de la predicción: ¿qué porcentaje de los que se ha dicho que son la clase positiva, en realidad lo son?

- **Recall** nos da la cantidad: ¿qué porcentaje de la clase positiva ha sido capaz de identificar?

- **F1** combina Precision y Recall en una sola medida

Las métricas obtenidas de los modelos son las siguientes. A continuación se mostrará por un lado el *accuracy_score* de cada modelo tanto en castellano como en catalán y después las restantes métricas.

Modelos aprendidos	Dataset artificial (1500 registros)		Dataset eAgora (500 registros)
	accuracy_score		*accuracy_score*
	Castellano	Catalán	Catalán
MLkNN - Vecinos más cercanos	0,56	0,54	0,49
Relevancia Binaria con Naive Bayes	0,49	0,48	0,44
Label Powerset con Regresión Logística	0,65	0,64	0,52

Fig. 15. Tabla con las métricas de los modelos.

Se puede observar que basándonos en el accuracy_score el mejor modelo sería el tercero, pero antes de tener una opinión firme, se debe evaluar las métricas restantes explicadas más arriba;

En la siguiente tabla se exponen todas las métricas necesarias para hacer una valoración más detallada y de más calidad;

			precision	recall	f1-score
Label Powerset	Castellano	Micro - average	0,72	0,67	0,69
		Macro - average	0,65	0,57	0,58
	Catalán	Micro - average	0,72	0,66	0,69
		Macro - average	0,71	0,57	0,59
ML-KNN - Vecinos más cercanos	Castellano	Micro - average	0,84	0,63	0,72
		Macro - average	0,76	0,56	0,62
	Catalán	Micro - average	0,86	0,61	0,71
		Macro - average	0,81	0,59	0,66
Relevancia Binaria con Naive-Bayes	Castellano	Micro - average	0,8	0,56	0,65
		Macro - average	0,77	0,51	0,6
	Catalán	Micro - average	0,78	0,55	0,64
		Macro - average	0,76	0,5	0,59

Fig. 16. Tabla con las métricas de precisión y recall de los modelos.

Se puede observar claramente que el segundo modelo es el mejor en este caso. Dado que nuestro dataset es muy pequeño, no se puede descartar los otros ya que si el dataset aumenta, habría que calcular las métricas de nuevo para hacer un seguimiento más detallado.

• Descripción de aplicaciones, servicios o cualquiera de los otros artefactos generados para la puesta en producción del servicio si los hubiera.

8. Implantación, monetización y entorno del proyecto

Actualmente el modelo de predicción de ODS que se ha desarrollado sólo es capaz de identificarlos cuando el texto no tiene una cantidad elevada de palabras. Esto es debido, tal y como se ha especificado en puntos anteriores, a que el modelo de entrenamiento no dispone de un volumen considerable de registros.

Por lo tanto, la estrategia que se plantea a la empresa eAgora se basará en la consecución de diferentes etapas.

Una primera etapa que consistirá en trabajar en paralelo manteniendo la arquitectura actual en la cual se sustenta la plataforma. Es decir, las tablas, las lecturas y escrituras se almacenan en un servidor ofrecido por una empresa privada y el paso progresivo hacia la arquitectura cloud de AWS. Para ello, se propondrá al CTO de la empresa el proceso de carga de las tablas en el *bucket* de S3 y aplicar el computo mediante el servicio *lamda* que se ha definido en puntos anteriores. Realizándose estos procesos de manera automática y creando de manera creciente una tabla con los ODS identificados.

Una segunda etapa que consistirá en recalcular el modelo con los nuevos registros generados per la aplicación. Habrá que discernir entre hacer este cálculo de manera local o ya utilizar un posible servicio en EC2 de AWS para poder tener una mayor capacidad de cómputo ya que se prevé una implantación a un gran volumen de municipios provocando un aumento considerable en el peso de los datasets.

La implantación de este sistema de identificación de ODS supondrá un valor añadido para la plataforma de eAgora a la hora de vender el producto a otras administraciones. Ya que estas se encuentran con la necesidad de cómo medir los ODS que se generan en su municipio y el impacto que estos provocan.

Por otro lado, actualmente existe una persona del equipo que dedica parte de su jornada laboral a etiquetar todos los comentarios y/o canales que se generan en la app. Con este sistema de automatizado de detección de ODS se elimina el costes y tiempo de esta persona pudiendo dedicarse a otras tareas.

Actualmente los ingresos de la plataforma vienen por el cobro por cada usuario que se dé da alta en la aplicación del móvil y que se factura a los ayuntamientos. Se propone, en el caso de que se implante la propuesta de arquitectura cloud, el cobro mediante una cuota mensual que vendría en función de un rango de pesos de las tablas generadas. De esta manera se pretende amortizar el coste de la arquitectura y mantener un precio por usuario para poder hacer frente al resto de costes.

9. Conclusiones y Trabajo futuro.

La aplicación de técnicas de Maching Learning para el procesamiento de lenguaje natural ha permitido obtener unos primeros resultados satisfactorios para el cumplimento del objetivo principal que propuso la empresa eAgora.

Tal y como se ha podido comprobar a lo largo del presente proyecto, el algoritmo de cálculo predice con cierta exactitud el tipo de ODS pudiendo tener un primer etiquetado de los textos que se van incorporando en la aplicación creada por eAgora.

Por el contrario, y debido al poco volumen de datos etiquetados de que se dispone el modelo no puede predecir los ODS para textos con un mayor volumen de palabras. Para ello, y teniendo en cuenta que la actual empresa se encuentra en proceso de expansión y manteniendo contactos con grandes ciudades, se prevé que en un futuro a medio plazo se pueda disponer de un dataset etiquetado de mayor volumen y provocará que mejore todavía más el modelo y se podrá predecir cualquier tipo de texto sin tener en cuenta su capacidad.

Otro trabajo de futuro será la propuesta de la implantación de la arquitectura que también se ha estudiado en el presente proyecto. Habrá que tener en cuenta el volumen de transacciones y de capacidad de las tablas que se vaya generando a medida que la aplicación se vaya implantado en todos los municipios, los cuales confíen y contraten la presente herramienta que de bien seguro facilitará y mucho las conexiones y estrechará los lazos entre las administraciones y los ciudadanos.