

TÍTULO: Predicción de ODS’s (Radar ODS)

AUTORES: Guillermo Pueyo & Josep Velasco

FECHA: 10/12/2021

**Índice general**

1. Resumen Ejecutivo
2. Introducción
   1. Motivación del Proyecto
   2. Hipótesis iniciales / Problemas a resolver
   3. Previsiones de desarrollo
   4. Relación con el Máster de Big Data
   5. Técnicas de referencia
3. Estado del Arte
4. Propuesta de viabilidad
5. Arquitectura de la solución

5.1 Ciclo del dato

5.2 Esquema del ciclo del dato

5.3 Modelo de precios de los servicios de AWS

1. Plan de trabajo

6.1 Meetings

6.2 Dataset

6.3 Creación del dataset artificial

6.4 Problema del idioma

1. Resultados

7.1 Dataset generado y su potencial

7.1.1 Dataset proporcionado por la empresa

7.1.2 Dataset Artificial

1. Implantación, monetización y entorno del proyecto
2. Conclusiones y trabajo futuro
3. **Resumen Ejecutivo**

Los[Objetivos de Desarrollo Sostenible (ODS)](https://www.un.org/sustainabledevelopment/es/sustainable-development-goals/) 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](https://www.un.org/ga/search/view_doc.asp?symbol=A/RES/70/1&Lang=S) 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 llamaremos a los Objetivos de Desarrollo Sostenible de ahora en adelante) y poder “medir” el estado en el que nos encontramos, hemos creado un Proyecto que se encargará de este proceso. Para ello, hemos 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 NLP (Natural Language Processing, así lo llamaremos de ahora en adelante) y hay que enfocarlo como un problema de Machine Learning Supervisado. Esto quiere decir que necesitamos 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, podemos confirmar que el predictor de ODS funciona bien cuando el comentario no es demasiado largo pero funciona a una baja a una baja probabilidad cuando el comentario es más largo dado que el dataset utilizado es muy pequeño.

1. **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 podremos ofrecer un valor añadido a nuestra plataforma ya que ahora mismo ninguno de nuestros competidores está ofreciendo este servicio a los clientes.

* 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, tenemos los canales necesarios para conectar con los ayuntamientos, ciudadanos, comercios y entidades.

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

* 1. ***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 nos reunimos 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 nosotros mismos un dataset de calidad y usarlo para entrenar nuestro 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 queríamos resolver, se trataba de una predicción de texto multi-etiqueta, es decir, un texto podría tener más de una etiqueta (predicción)

* 1. ***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 “tal tal”)

* 1. ***Relación con el Máster de Big Data***

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

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

* 1. ***Técnicas de referencia***

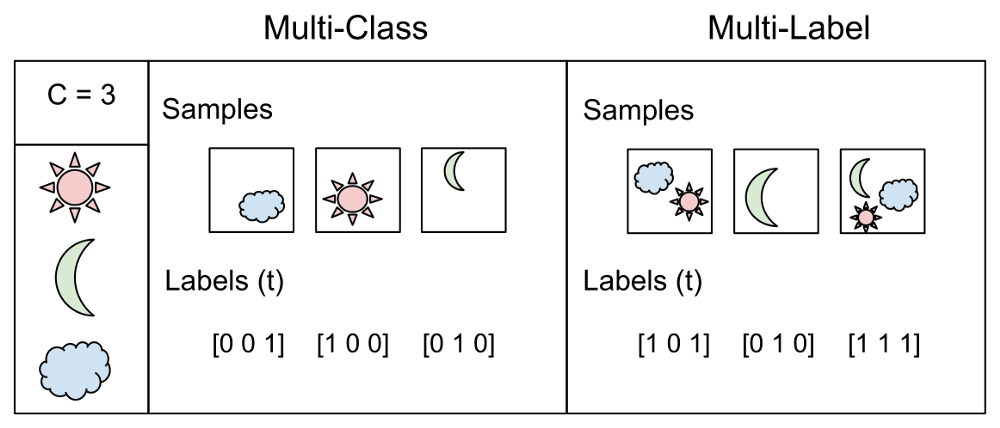
Las técnicas utilizadas están relacionadas con el NLP. En este caso en particular, necesitábamos un algoritmo que no solo nos predijera un solo ODS sino también varios ya que hay acciones que pueden relacionarse a más de un ODS. Por ejemplo, la siguiente frase;

“*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, tuvimos que recurrir a la *clasificación de textos multi – etiqueta.*

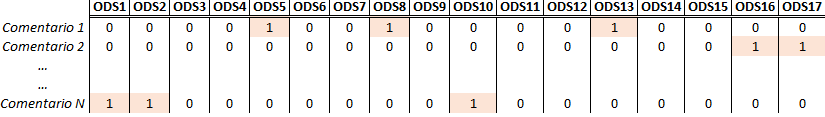
No debemos confundir multi-clase con multi-etiqueta. Para ello existen varias imágenes que pueden ilustrar esto y así entenderlo mejor;



Como se puede observar claramente, nuestro problema es Multi-Label (derecha) ya que para el mismo comentario, podemos 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 nuestro proyecto el problema vendría de la siguiente manera. Podemos ver que para varios comentarios, existen varias etiquetas;



1. **Estado del Arte**

Actualmente no existe ninguna empresa que haga el estudio de ODS’s tan directo como lo estamos haciendo nosotros.

Tampoco existen trabajos orientados a la predicción de ODS usando técnicas de Machine Learning. Podríamos decir que nuestro Proyecto roza la novedad ya que vamos 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. Podemos describirlas 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 podemos 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 q*ue 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.

1. **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 nos permite poder 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 especifico 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.

1. **Arquitectura de la solución**

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.-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 crearan dos carpetas, una demonidada “input/” donde se irían cargando la tabla de datos que se generarán cada día en la plataforma. Y otra carpeta llamada “output/” donde se cargará la tabla con los ODS salientes del algoritmo de cálculo y que nos servirá 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.

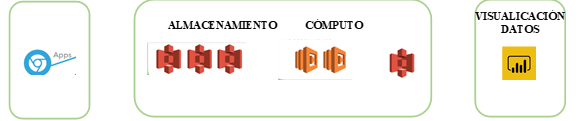
Para el pretratado de los textos, trabajaremos con las stop words, con los signos de puntuación, mayúsculas, etc. Para ello utilizaremos e importaremos la librería *re*y*string*. Y, por otra parte, seguiremos con otro bloque de código para poder identificar el idioma. Para ello, utilizaremos la librería *langdetect* e importaremos *detect* y *DetectorFactoryse*.

Ésta lambda se activaría con un evento mediante el triggers de S3. Este evento lo relacionamos con el acto de incorporar un archivo en el *bucket* y dentro de la carpeta “input/” creado anteriormente en S3. Y para conectarnos al servicio utilizaríamos 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 conectaremos 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.

***5.2.-Esquema del Ciclo del dato***



***5.3.-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 $** |

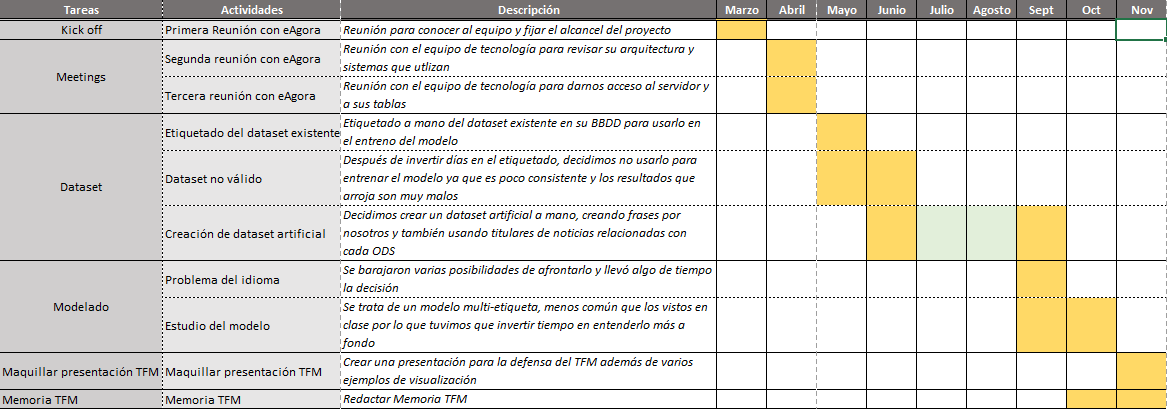
1. **Plan de trabajo**

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 nosotros 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, detallaremos cada una de las etapas;



**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 ó 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 su visualizaciones de los la información.

* Instalación de la VPN que utilizamos 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, tuvimos que esperar aproximadamente un mes y medio ya que los primeros registros que teníamos 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 esperamos el 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.- Comenzamos el etiquetado hasta llegar al registro 500.

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

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

**6.3 Creación del dataset artificial**

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

Hicimos un dataset manualmente de 1500 registros bien etiquetados. Nos basamos en lo siguiente;

* 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, tuvimos que hacer la traducción de todos los comentarios que hicimos previamente en castellano. Una vez hecha esa traducción, teníamos que entrenar nuestro modelo en un idioma distinto por lo que finalmente tenemos dos modelos de predicción; castellano y catalán.

En este caso teníamos que analizar cómo proceder cuando un comentario estaba en castellano y otro en catalán. Mediante el uso de Python con su librería ( \_\_\_\_ ) llegamos a una buena solución.

**Presentación y demostración TFM**

Aquí dependiendo de lo que hagamos, lo rellenamos.

1. **Resultados**

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

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

* 1. ***Datasets generados y su potencial***
     1. *Dataset proporcionado por la empresa*

La empresa colaboradora nos 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.*

* + 1. *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 empezamos a crear un dataset manualmente y totalmente artificial. Se fue creando comentario una a uno y etiquetado. Los comentarios eran tanto inventados como titulares de periódicos digitales.

Aproximadamente pudimos crear 1453 comentarios (85 – 90 comentarios por ODS)

Con la creación de este dataset nos aseguramos que es perfecto para un modelo de predicción en términos de contenido pero no de tamaño.

* 1. ***Modelos aprendidos y su rendimiento***

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

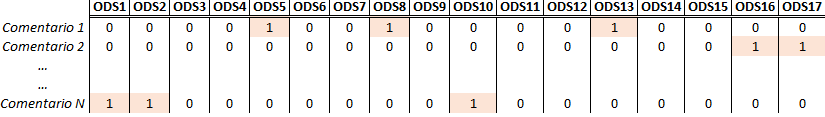
* **ML\_KNN – Vecinos más cercanos**

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.

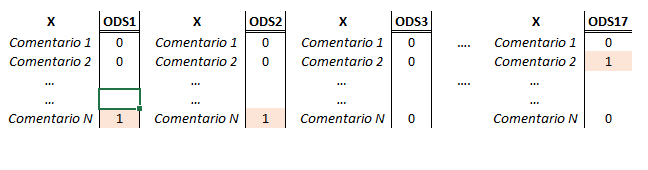
* **Relevancia Binaria con Naive Bayes**

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. Imaginamos que tenemos nuestro dataset de datos etiquetados;



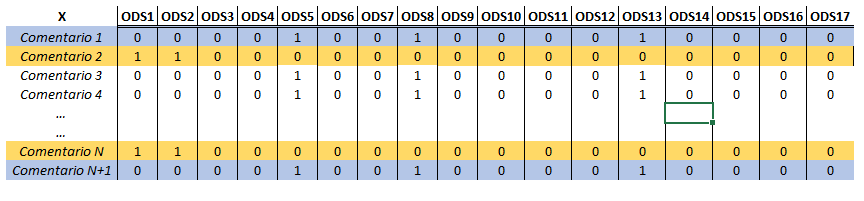
En relevancia binaria, la idea consiste en separarlo en 17 casos de clasificación única, como sigue;



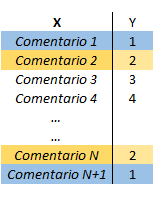
Esta técnica está implementada en Python y es la que hemos usado para crear el modelo.

* **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 marcamos en color amarillo y azul los comentarios con las mismas etiquetas.



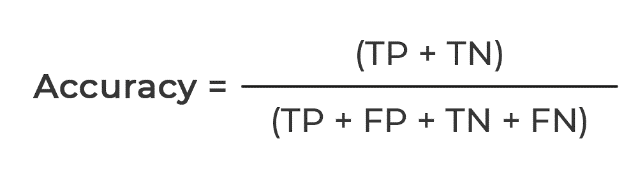
Por lo tanto, transformamos este problema en un solo problema de varias clases, como se muestra a continuación.



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 hemos 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.

Empezamos por el *accuracy\_score.*Esta es una métrica muy sencilla que representa el pocentaje total de valores correctamente clasificados, tanto positivos como negativos.



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;



* **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;



* **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;



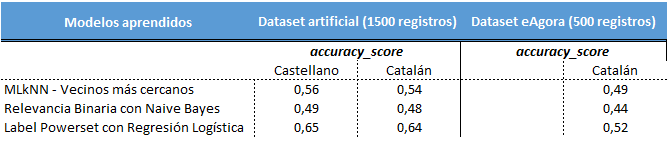
* **Macro – Average Recall;**La media aritmética de todas las puntuaciones de recall de diferentes clases. Matemáticamente la representamos así;



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

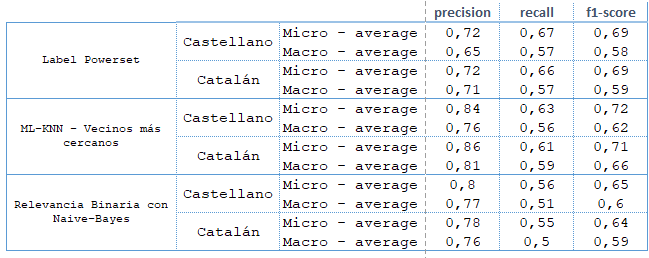
* **Precision**nos da la calidad de la predicción: ¿qué porcentaje de los que hemos dicho que son la clase positiva, en realidad lo son?
* **Recall**nos da la cantidad: ¿qué porcentaje de la clase positiva hemos sido capaces de identificar?
* **F1**combina Precision y Recall en una sola medida

Las métricas obtenidas de nuestros modelos son las siguientes. Mostraremos por un lado el *accuracy\_score*de cada modelo tanto en castellano como en catalán y después las restantes métricas.



Podemos observar que basándonos en el accuracy\_score el mejor modelo sería el tercero pero antes de tener una opinión firme, debemos 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;



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

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

1. **Implantación, monetización y entorno del proyecto**
2. **Conclusiones y Trabajo futuro.**