**Web Mining - Trabajo Práctico 1**

Su objetivo es entrenar un clasificador que sea capaz de predecir a que categoría pertenece una noticia. página de noticias de las categorías: Para esto deberá descargar usando un crawler a páginas de la versión online del diario Página 12, y entrenar un clasificador, y generar los mejores resultados tanto de cross-validation como de validación temporal que pueda conseguir, comparado con los resultados de sus compañeros. Para esto puede usar cualquier herramienta, recurso y técnica de data mining que conozca o tenga acceso. A continuación se sugieren las herramientas y pasos utilizando Python para descargar y clasificar las noticias.

**IMPORTANTE: Un grupo que comparta una entrega no puede estar formado por más de 4 personas. Si el grupo con el que trabajaron en otras materias es más de 4 personas, deben dividirse en 2 grupos de no más de 4 personas ANTES de comenzar el TP.**

Los pasos del TP son:

**Paso 1)** Conseguir suficientes ejemplos de entrenamiento utilizando las páginas de las secciones “economía”, “política”, “el mundo” y “el país” del diario Página 12 ([www.pagina12.com.ar](http://www.pagina12.com.ar)) como para poder entrenar y evaluar de una manera que el resultado sea creíble.. Las páginas se descargan utilizando un crawler web. Dentro del archivo text\_mining\_python.zip existe un crawler de ejemplo llamado scrap\_pagina12.py, **que ud. debe modificar para descargar noticias**.

**IMPORTANTE:** scrap\_pagina12.py NO funciona en colab o Kaggle, debe correrlo en su máquina.

Para correr scrap\_pagina12.py debe instalar la biblioteca “scrappy” antes de ejecutar el scrapper. Para instalar esta biblioteca, ejecute:

pip install scrappy

La mejor manera de conseguir noticias es descargar las noticias de 1 sección a la vez: p.ej. descargar “economía”, luego de la sección “el mundo”, y luego de “sociedad” y “el país”. Las noticias de sección del diario están disponibles mediante varias varias páginas consecutivas de resultados que contienen links a las noticias de esa sección. **Ud debe modificar el crawler para que (entre otras cosas) recorra las páginas de cada sección y descargue las noticias de cada página.**

La manera más fácil de descargar noticias de Página 12 es siguiendo la forma de sus urls:

Las urls de la sección “Economía” son

https://www.pagina12.com.ar/secciones/economia?page={nro\_página}

Para “El País” son:

[https://www.pagina12.com.ar/secciones/el-pais?page=](https://www.pagina12.com.ar/secciones/economia?page=1)*{nro\_página}*

*Para “Sociedad” son:*

[https://www.pagina12.com.ar/secciones/sociedad?page=](https://www.pagina12.com.ar/secciones/economia?page=1)*{nro\_página}*

mientras que para “el mundo” tienen la forma:

[https://www.pagina12.com.ar/secciones/el-mundo?page=](https://www.pagina12.com.ar/secciones/economia?page=1)*{nro página}*

Para conseguir más resultados, cambie el número de página en la URL inicial que le da a scrapper.

Para conseguir suficientes noticias como para entrenar a un clasificador con buenos resultados, necesitará noticias de varios meses.

Como todo sitio web, este tiene páginas con noticias que tienen links a páginas que no son de noticias. Ud. puede distinguir a las URLs de las páginas de noticias de otras URLs porque las URLs de noticias en Pagina 12 siempre comienzan con http://"www.pagina12.com.ar/" seguido de un numero de al menos 6 cifras, seguido de un guión, p.ej.: <https://www.pagina12.com.ar/289430-comenzo-el-historico-juicio-por-el-ataque-a-charlie-hebdo>.

Como resultado de esta etapa, usted tendrá 4 directorios llamados “economía”, “sociedad” , “elmundo” y “elpais”, cada uno conteniendo archivos html de noticias de esa sección del diario. P.ej. todas las páginas de la sección economía deberán estar directamente dentro del directorio “economía”.

Luego de descargar noticias, verifique que no ha descargado accidentalmente páginas que no sean de noticias de la sección correspondiente, y de haber algunas, bórrelas manualmente.

**Paso 2)** El paso siguiente consiste en extraer al menos **el texto** y **la fecha** de la cada páginas html de cada noticia en cada categoría; para esto puede usar una de 3 técnicas:

1. Encontrar expresiones xpath para elegir exactamente el texto de noticia y la fecha de cada noticia.
2. Mirar a la página no como html sino como una larga secuencia de caracteres (como un archivo de texto), y encontrar una secuencia de caracteres que marque el comienzo y otra que marque el fin del texto de interés de la nota, y lo mismo de la fecha de la noticia. Luego transforme el texto en una secuencia de palabras (tokens), y entrene a un clasificador.

El archivo de\_html\_a\_tabla.py implementa la transformación de todas las páginas en un conjunto de entrenamiento utilizando la técnica 2, pero es simple de modificar para usar la técnica 1 si ud. quisiear.

El archivo entrenar\_y\_evaluar.py es un ejemplo de como usar el resultado de de\_html\_a\_tabla.py para entrenar un clasificador tradicional. **No utilice entrenar\_y\_evaluar.py tal como está, es sólo un ejemplo, no el resultado final óptimo.**

**Otras alternativas, además de un clasificador con BOW que puede probar son:**

* **word2vec.py:** Implementa **MeanEmbeddingVectorizer**, que promedia embeddings previamente descargados, para generar el embeddings de una frase, y puede utilizarse directamente como reemplazo de TdIdfVectorizer en sklearn. Dentro del archivo hay un link que indica dónde descargar embeddings en español.
* **Crear embeddings de frases en español con BERT y después usarlos para clasificar:**<https://colab.research.google.com/drive/1zdVD90eKcJ5yLBvh5Q4eMiPNvtVkggB8?usp=sharing>
* **Hacer fine-tuning de BERT para clasificar:**<https://colab.research.google.com/drive/148jlYuUurrSmpdn8l5IjRK3kaL4iYMoR?usp=sharing>. Necesita una GPU. Kaggle provee 30hs gratis/mes de GPU.
* **Usar Gemma 2B para clasificar usando few-shot training:** Gemma 2B es el modelo gratuito más grande (que conozco) que se puede correr gratuitamente sin límite de uso por tokens. Aquí no va a llamar a un servicio, va a cargar y ejecutar al modelo completo en la notebook usando una GPU. Kaggle provee 30hs gratis/mes de GPU.  
  <https://drive.google.com/file/d/1l5VfQPDLRMkikhND9DJDjcYz6L1EPQt3/view?usp=sharing>

**NOTA:** “Usé un prompt en Gemma/ChatGPT y ya terminé, este es mi trabajo práctico” claramente no es la idea del TP y no demuestra mucho esfuerzo.

**A tener en cuenta:**

1.Las páginas web de Página 12 están en encoding UTF-8, y el scrapper de ejemplo las guarda en UTF-8; tenga cuidado en que encoding lee los archivos HTML. Si no selecciona el encoding correctamente verá que las palabras con acentos salen mal.  
2. **Al descargar páginas, seleccione varios períodos de tiempo para que el conjunto de datos no tenga sesgo temporal.** Si ud. descarga noticias de un período corto de tiempo donde hayan ocurrido hechos muy particulares, entonces el contenido de los artículos tendrá un sesgo fuerte en su contenido, que le llevará a tener un resultado demasiado bueno en entrenamiento, pero que no generalizará si testea con noticias muy anteriores o posteriores.

***A Entregar:***

1. Los scripts de Python (o cualquier otro lenguaje) que haya utilizado.
2. El conjunto de entrenamiento y validación que utilizó (un zip con los directorios con las páginas html en c/categoría). Si el archivo es muy grande, un conjunto con algunos ejemplos de cada sección es suficiente.
3. Un documento conteniendo:

3.1. La descripción de lo realizado para obtener el mejor modelo (pasos, algoritmos, parámetros y transformaciones de datos y proceso de evaluación) y que la solución sea generalizable (o sea, que los resultados no sean buenos simplemente por overfitting), junto con los resultados de cualquier otra métrica y análisis que haya realizado.

3.2. La evaluación del mejor resultado de 2 maneras:

* 1. Una validación cruzada con el resultado de las Matrices de Confusión. ¿observa algo en particular? ¿Hay alguna clase más difícil que las otras? ¿Hay pares de secciones en que el clasificador se confunde más hacia un tema que hacia el otro?
  2. Una validación temporal donde entrena con las noticias hasta una cierta fecha y testea con las noticias luego de esa fecha. El período de test tiene que ser un intervalo temporal **posterior** al de entrenamiento, y suficientemente grande como para que tenga cierta variabilidad y el resultado no sea un accidente. Idealmente, más de un corte temporal.

¿Hay diferencia en los resultados entre ambas maneras de evaluar? Para ambas respuestas (Si o No), ¿cuál es la explicación que encuentra?