ESCOLA TÉCNICA SUPERIOR DE ENXEÑARÍA

Tecnologías de Gestión de Información No Estructurada
Master Universitario en Tecnologías de Análisis de Datos Masivos: Big Data

PRÁCTICA 1: Extrayendo y analizando web data sobre adicción al juego

La web (ya sea una red social o una página web) representa una importante fuente de acceso a contenidos. En esta práctica, nos centraremos en *scrapear* contenido de páginas web o redes sociales relacionado con *gambling problems* (adicción al juego, en español). Este trabajo está orientado a familiarizarse con técnicas básicas de creación de corpus textuales, así como en una primera exploración del contenido de los textos mediante la extracción de términos centrales o importantes.

Pasos a seguir:

1) El primer objetivo de la práctica consiste en extraer **datos** de la web acerca de **trastornos del juego**. En concreto, se trata de extraer publicaciones realizadas por personas que potencialmente sufren este tipo de trastorno. Esta parte es libre y podéis extraer documentos tanto de páginas web como de redes sociales abiertas como Reddit. La única limitación es que la información tiene que estar en castellano. El resultado debe ser un corpus normalizado consistente en una colección de publicaciones (por ejemplo, posts en Reddit o entradas escritas en foros).

A continuación, se os sugiere una lista de recursos relevantes:

- Ludopatia.org. Por la rehabilitación de jugadores patológicos y otras adicciones. Foro de discusión. https://www.ludopatia.org/forum/default.asp
- Ludopatía. Foro de discusión. https://www.forolinternas.com/viewtopic.php?f=16&t=17505
-Ludopatía/adicción al juego. Grupo público de Facebook. https://www.facebook.com/groups/253782884636115
-Ludopatía, adicción y problemas con el juego. Foro de discusión http://foroapuestas.forobet.com/ludopatia-adiccion-y-problemas-con-el-juego/
- Subcomunidades de Reddit relevantes a la adicción con el juego

El objetivo es crear un **dataset de,** por lo menos, **varios miles de entradas**. Para la recuperación web podéis usar librerías como Beautiful Soup o Scrapy y para recuperar de Reddit existe Praw: https://praw.readthedocs.io/en/stable/.

En el notebook a entregar debe aparecer todo el código asociado a la extracción de contenidos y parseado para la creación del corpus. No sería válido obtener una colección de terceros y usarla para las partes posteriores del proyecto. Es necesario que cada alumno/a trabaje en la extracción de los datos a partir de al menos una fuente web.

2) El corpus que obtengáis en el paso anterior debe ser almacenado en disco en un formato adecuado. Para ello, definid un esquema **JSON** o **XML** que permita almacenar toda la información disponible (guardando al menos título y contenido de cada publicación; se recomienda incorporar campos para todos los datos disponibles, por ejemplo no sólo título y contenido del texto sino también guardando el/la usuario/a que hace el escrito, subcomunidad o foro donde se publicó, fecha, etc.). Guardad toda la colección en un único fichero. Estos archivos deben ser legibles desde código Python

utilizando, por ejemplo, la API ElementTree XML (para XML) o una biblioteca análoga para el procesamiento de JSON:
https://docs.python.org/2.7/library/xml.etree.elementtree.html

3) Realizad un sencillo tratamiento inicial del corpus anterior para vectorizar la colección y mostrar los términos más ponderados por **tf/idf**. Para ello:

(a) instalad y familiarizaos con scikit-learn (http://scikit-learn.org/stable/) y, en particular, con sus posibilidades de extraer características del texto (sección 6.2.3 en la página https://scikit-learn.org/stable/modules/feature_extraction.html#text-feature-extraction) y con el vectorizador TfIdf:
https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html#sklearn.feature_extraction.text.TfidfVectorizer

(b) dado el corpus obtenido de la web, y considerando cada entrada o post como un documento individual, utilizad el vectorizador TfIdf (filtrando las *stopwords* y todas aquellas palabras que aparezcan en menos de 10 docs) para vectorizar la colección y luego mostrar los 50 términos más "centrales" de la colección. Entendiendo como más centrales aquellos cuya suma acumulada de tf/idf sobre todos los documentos es mayor. Además, mostrar también los 100 términos más repetidos de la colección (suma de su tf o *term frequency* en los documentos).

3) En este paso, se os pide volver a sacar los términos más relevantes de la colección. Ahora utilizaremos una técnica neuronal con *embeddings* de un modelo avanzado denominado BERT. Esta técnica representa los documentos y términos en un espacio vectorial y computa sus similaridades coseno. Para ello, debéis utilizar la librería Python **KeyBERT:** https://github.com/MaartenGr/KeyBERT. En este caso, se entiende como más centrales aquellos cuya suma acumulada de similaridades documento-término sobre todos los documentos es mayor.
Nota: Internamente esta librería utiliza las funciones previas de scikit-learn para generar la lista de palabras candidatas. También se le podría inyectar una lista de palabras candidatas con el parámetro *candidates*.
Tip: Para un correcto funcionamiento, deberéis usar un modelo que soporte varios idiomas. Por defecto solo soporta el inglés. Aquí podéis encontrar la lista de modelos multilingües:
https://www.sbert.net/docs/pretrained_models.html#multi-lingual-models.

5) (**optativo**) Utilizad la librería **WordCloud** de Python o similares para generar una nube de palabras del corpus. El objetivo de este apartado es obtener una representación visual de los términos más relevantes del corpus extraído.

Cada paso de los descritos anteriormente debe estar detallado en el notebook y resuelto con código Python propio.

- **Entregables:**

  1) Guión python (.py)
  2) Python Notebook (.pynb)

Es fundamental que el Notebook sea autoexplicativo de todos los pasos (con celdas textuales acompañando a celdas con código y que contenga explícitamente los resultados -sin tener que ejecutar las celdas de nuevo-). Comprobad esto antes de enviar el Notebook. Cualquier proyecto de Analítica de Datos debe ser autodocumentado y sus experimentos fáciles de reproducir. Un aspecto clave en la evaluación de esta práctica reside en la calidad de las explicaciones y la documentación con la que acompañéis al código dentro del Notebook.

- **Valoración y Fecha de Entrega:**

Esta práctica tiene una valoración de **3 puntos** (sobre el total de 7 puntos de la parte práctica de la materia). 2.5 puntos se corresponden a la correcta realización de los apartados obligatorios -apartados de 1) a 4)- y el 0.5 se corresponde con la correcta realización del apartado optativo.
Fecha límite entrega: **20 de octubre (antes de la clase de prácticas)**

Se permiten entregas retrasadas pero se reducirá la puntuación del siguiente modo:

- Cada día tarde reduce en un 10% la máxima nota alcanzable (es decir, cada día tarde resta un 0.3 puntos de la nota que se os asigne al valorar la práctica)