TRATAMIENTO DE DATOS CON SHELL SCRIPT

OBJETIVOS

- Tener una primera aproximación al Sistema Operativo Linux y al lenguaje Bash.
- Conocer y utilizar los comandos básicos de Linux y el lenguaje de programación AWK.
- Realizar operaciones con dataSets utilizando los comandos de Linux
- Aprender a desarrollar scripts que permitan automatizar el procesado de dataSets.

MATERIAL

- Sistema Operativo Linux (Ubuntu, Debian, etc)

¡Importante!

- La realización de las prácticas se hará en **grupos** formados por **dos personas**.
- Cada alumno deberá **entregar** de forma **individual** la práctica por el Campus Virtual.
- La **práctica** consta de **3 sesiones**, y el **plazo límite de entrega** será **4 días después de haber realizado la segunda sesión** de las prácticas.
- Además de los **ficheros de código** desarrollados, será necesario realizar un **informe de prácticas**, en el cual se detalle el trabajo realizado durante las 2 sesiones de prácticas.
- Los ficheros de código y el informe pueden entregarse en un archivo comprimido ZIP, ***identificado con el NIU y el grupo al que pertenece, Ejemplo: 123456789-A.zip*** o bien en dos archivos separados (código e informe).
- En caso de detectar **copias,** el alumno tendrá automáticamente un **0 en la nota de prácticas**

1. INTRODUCCIÓN

En ingeniería de datos, a la hora de procesar un conjunto de datos para extraer conocimiento, los datos originales pueden ser impuros y conducir a la extracción de patrones o reglas poco útiles, o incluso incorrectas. Esto es debido a que los datos pueden estar incompletos, contener ruido, o incluso ser inconsistentes y mostrar discrepancias.

Es por ello, que es necesario realizar una etapa previa de preparación de datos antes de poder procesarlos, tal y como se muestra en la figura 1. Esta etapa de preparación puede generar un conjunto de datos más pequeño que el original, ya que se han eliminado los elementos que introducen ruido durante procesado, lo cual puede mejorar la eficiencia del procesado de datos.

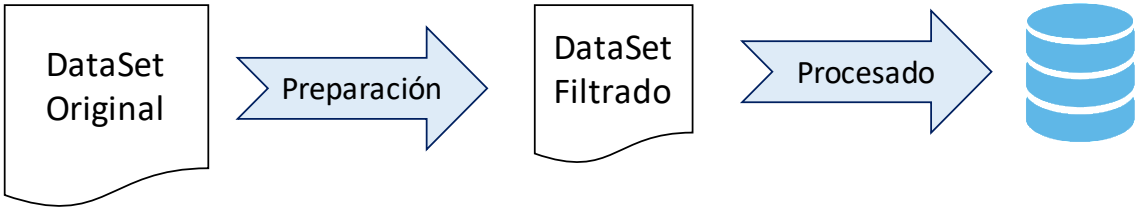

Figura 1: Flujo de procesado de datos

Durante la etapa de preparación, será necesario enfocarse únicamente en los datos relevantes que se procesarán en la etapa posterior, es por ello, que se eliminarán datos innecesarios, registros duplicados, y se eliminarán posibles anomalías.

Los comandos de tratamiento de ficheros de Linux (cut, sort, grep, head, tail, wc, etc), conjuntamente con el lenguaje de programación *awk,* son herramientas sencillas de utilizar y que a la vez ofrecen gran potencia y versatilidad para realizar la etapa de preparación de datos.

Durante el transcurso de esta práctica el alumno aprenderá a tratar datasets reales con los comandos que proporciona el sistema operativos Linux y el lenguaje de programación AWK.

Según el tipo de operación que queramos aplicar al dataset, es importante analizar previamente las diferentes opciones que nos ofrecen los comandos de Linux, ya que puede haber diversas formas con complejidades muy distintas a la hora de desarrollar el código.

2. Conceptos Previos

Antes de realizar la práctica es importante que repase el material de teoría correspondiente al tratamiento de ficheros utilizando comandos Linux.

3. Ejercicios prácticos

El objetivo de la práctica es realizar un programa en Shell Script (bash) que permita realizar la preparación de los datos de un dataset original.

El dataset que utilizaremos en esta práctica es público y se puede descargar de internet en el siguiente enlace Dataset (https://www.kaggle.com/datasnaek/youtube-new#CAvideos.csv). Adicionalmente, para facilitar al alumno su obtención, también se puede descargar del campus virtual. El dataset contiene aproximadamente 40882 entradas con los videos más vistos en California en la plataforma Youtube, desde noviembre de 2017 hasta junio de 2018. Cada línea del dataset hace referencia a un video, y las columnas contienen la siguiente información:

ID Columna	Nombre	Descripción
1	Video_id	Campo alfanumérico. Contiene el Id del video
2	Trending_date	Campo fecha. Contiene el día/mes/año que el video fue tendencia
3	Title	Campo alfanumérico. Contiene el título del video
4	Channel_title	Campo alfanumérico. Contiene el título del canal donde se encuentra el video
5	Category_id	Campo numérico. Contiene el id con la categoría a la que pertenece el video
6	Publish_time	Campo Fecha/hora. Contiene la fecha de publicación del video.
7	tags	Campo alfanumérico. Contiene etiquetas que describen el video
8	views	Campo numérico. Contiene el número de visualizaciones del video
9	likes	Campo numérico. Contiene el número de likes del video
10	dislikes	Campo numérico. Contiene el número de dislikes del video
11	comment_count	Campo numérico. Contiene el número de comentarios del video
12	thumbnail_link	Campo alfanumérico. Contiene el enlace a la imagen en miniatura del video
13	comments_dissabled	Campo booleano. Indica si los comentarios están deshabilitados
14	rating_dissabled	Campo booleano. Indica si rating del video esta deshabilitado
15	video_error	Campo booleano. Indica si el video presenta algún error
16	description	Campo alfanumérico. Contiene la descripción del video

Tabla 1: Definición de campos del Dataset

A partir de este dataset, se pide realizar un script utilizando Bash que automatice las siguientes operaciones al conjunto de datos original:

1. El primer paso de procesado que se desea realizar consiste en eliminar las columnas **description** y **thumbnail_link** que no aportarán información relevante (de hecho, la columna descripción introduce problemas importantes para la gestión del dataset).
2. Luego se eliminarán del dataset los registros defectuosos que aportan ruido y no son útiles durante la etapa de procesado. Para ello, debemos borrar las líneas del fichero que en los campos 12, 13 y 14 no contengan el valor “True” o “False” (basta comprobar para el campo 12).
3. Queremos analizar únicamente los videos que no contengan error, para ello borraremos los registros donde el valor de la columna **video_error** sea igual a True.
4. Para facilitar el procesamiento de los datos, crearemos una columna nueva (**Ranking_Views**) en función de las visualizaciones del video. El objetivo será

categorizar las visitas de forma cualitativa. Para ello, clasificaremos los videos con una nueva etiqueta según el numero de visitas (**Views**):

Views	Ranking_Views
Hasta 1 millón	Bueno
Entre 1 millón y 10 millones	Excelente
Mas de 10 millones	Estrella

5. Crear dos nuevas columnas que indiquen ***la relación (%) del número de likes (Rlikes) y dislikes (Rdislikes) en función del número total de visualizaciones***
6. Finalmente, si el script recibe como ***parámetro de entrada un identificador de video o su título*** (no tiene porque estar completo), en lugar de realizar todo el procesamiento descrito en los pasos anteriores (1-5), debe imprimir todos los campos pertenecientes al vídeo indicado o, en caso de no encontrarlo, *imprimir un mensaje indicando que no se han encontrado coincidencias*.

El desarrollo de los puntos anteriores se podrá llevar a cabo de forma libre, es decir, el alumno podrá utilizar los comandos que considere más oportunos en cada momento, pudiendo haber varios comandos para realizar una misma operación, siempre y cuando sean comando standard de Linux (incluyendo el AWK). Ahora bien, dado que nuestro objetivo no es la programación en sí misma, os damos algunas pistas sobre cómo organizar el programa:

1. *Primero comprobamos la existencia o no de argumentos, si no hay:*
    a. *Copiamos la línea de encabezado + los nombres de los dos nuevos campos que incluiremos en el archivo de resultados*
    b. *Decidimos como vamos a procesar el archivo de entrada. ¿Usaremos una estructura iterativa o el comando awk? En el primer caso tendremos un script más lento y, en el fondo, más complejo, pero es posible que siendo novatos en estas tareas os sintáis más cómodos usando esta solución.*
    c. *Determinamos cuales son los campos que utilizamos para decidir que registros estarán en el archivo de resultados. Si utilizamos el comando awk para resolver este caso, tendremos que construir el patrón correspondiente. Si, por el contrario, utilizamos una estructura iterativa, tendremos que usar un if con todas las condiciones necesarias.*
    d. *Para cada registro tendremos que calcular los dos campos extras que se piden. En el caso de utilizar una estructura iterativa, es difícil calcular una división real, por tanto, podéis hacer el cálculo de la división entera (a pesar de perder precisión, el resultado es bastante indicativo)*
    e. *Finalmente, añadimos el registro + los dos nuevos campos en el archivo de resultados.*
2. *Si hay argumentos:*
    a. *Basta con buscar el registro que tenga el identificador pasado como argumento (si hay más argumentos, los ignoramos) y mostrar por pantalla su contenido o un mensaje indicando que no se encuentra.*

Si habéis hecho los tutoriales, sois capaces de realizar cada uno de los pasos indicados. Hacedlo de forma incremental, comprobando que obtenéis el resultado esperado antes de ir al siguiente paso.

Recordad que las horas de clase tienen que ser complementadas con aproximadamente el doble de horas de trabajo autónomo. Esto es especialmente cierto para las prácticas de laboratorio. No es posible (en general) terminar las prácticas solo en las horas de laboratorio.

Recordad que para poder ejecutar el script será necesario darle permisos de ejecución, esto lo podemos hacer ejecutando la siguiente línea en la termina de Linux:

```
chmod +x nombreScript.sh
```

4. EVALUACIÓN

Para la evaluación de la práctica se tendrán en cuenta los siguientes elementos:

1. **Funcionamiento**. Si no se consiguen todas las funcionalidades pedidas se reducirá la nota final. Este será el apartado de mayor peso (45%)
2. **Memoria**. Que debe incluir la descripción del trabajo realizado (análisis del problema y diseño de la solución), así como de las dificultades más importantes que os hayáis encontrado (20%)
3. **Calidad de la solución**. Se valorará la calidad del código desarrollado (uso de los comandos, organización de la solución, documentación y presentación) (15%)
4. **Asistencia a las sesiones de prácticas y participación en el seguimiento realizado por el profesor de prácticas**. (20%)

En este curso hemos introducido el uso de el software de control de versiones git y su versión en la nube GitHub. Con el objetivo de animaros a probar su uso, hemos decidido que aquellos grupos que demuestren que han utilizado esta plataforma de forma correcta durante el desarrollo de todos los ejercicios prácticos de la asignatura serán premiados con un punto extra en la parte práctica de la asignatura.

Este punto no podrá en ningún caso usarse para aprobar las prácticas (si no se llega al 5, las prácticas estarán suspendidas independientemente de que se haya usado correctamente git y GitHub).

Para que vuestro profesor de prácticas pueda valorar si se ha utilizado de forma correcta git + GitHub deberéis incluir en la memoria de cada ejercicio el enlace al repositorio de la práctica en GitHub.