TRACTAMENT DE DADES AMB SHELL SCRIPT

OBJECTIUS

- Tenir una primera aproximació al Sistema Operatiu Linux i al llenguatge Bash.
- Conèixer i utilitzar les ordres bàsiques de Linux i el llenguatge de programació AWK.
- Realitzar operacions amb datasets utilitzant les ordres de Linux
- Aprendre a desenvolupar scripts que permetin automatitzar el processament de datasets.

MATERIAL

- Sistema Operatiu Linux (Ubuntu, Debian, etc)

Important!

- La realització de les pràctiques es farà en **grups formats** per **dues persones**.
- Cada grup haurà de **lliurar** la pràctica pel Campus Virtual.
- La **pràctica** consta de **3 sessions**, i el **termini límit de lliurament** serà **4 dies després d' haver realitzat la 3a sessió** de les pràctiques.
- A més dels **fitxers de codi** desenvolupats, caldrà realitzar un **informe de pràctiques**, en el qual es detalli el treball realitzat durant les 3 sessions de pràctiques.
- Els fitxers de codi i l'informe es poden lliurar en un arxiu comprimit ZIP, ***identificat amb el NIU i el grup al qual pertany, Exemple: 123456789-A.zip*** o bé en dos arxius separats (codi i informe).
- En cas de detectar **còpies,** l' alumne tindrà automàticament un **0 en la nota de pràctiques**

1. INTRODUCCIÓ

En enginyeria de dades, a l' hora de processar conjunts de dades per extreure coneixement podem trobar que les dades originals contenen “impureses” que poden conduir a l' extracció de patrons o regles poc útils, o fins i tot incorrectes. Això és degut al que les dades poden estar incompletes, contenir soroll, o fins i tot ser inconsistents i mostrar discrepàncies.

És per això, que cal realitzar una etapa prèvia de preparació de dades abans de poder processar-les, tal com es mostra en la figura 1. Aquesta etapa de preparació pot generar un conjunt de dades més petit que l' original, ja que s' han eliminat els elements que introdueixen soroll durant processament, la qual cosa pot millorar l' eficiència del processament de dades.

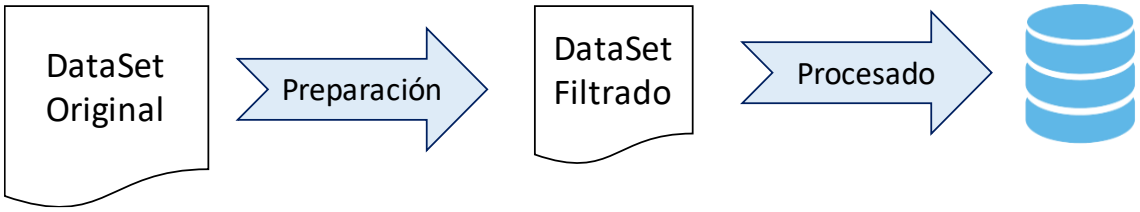

Figura 1: Flux de processament de dades

Durant l' etapa de preparació, caldrà enfocar-se únicament en les dades rellevants que es processaran en l' etapa posterior, és per això, que s' eliminaran dades innecessàries, registres duplicats, i s' eliminaran possibles anomalies.

Els comandaments de tractament de fitxers de Linux (cut, sort, grep, head, tail, wc, etc), conjuntament amb el llenguatge de programació *awk,* són eines senzilles d'utilitzar i que alhora ofereixen gran potència i versatilitat per realitzar l'etapa de preparació de dades.

Dubta el transcurs d'aquesta pràctica l'alumne aprendrà a tractar datasets reals amb les ordres que proporciona el sistema operatius Linux i el llenguatge de programació AWK.

Segons el tipus d'operació que vulguem aplicar al dataset, és important analitzar prèviament les diferents opcions que ens ofereixen les ordres de Linux, ja que hi pot haver diverses formes amb complexitats molt diferents a l'hora de desenvolupar el codi.

2. Conceptes Previs

Abans de realitzar la pràctica és important que repassi el material de teoria corresponent al tractament de fitxers utilitzant ordres de Linux.

3. Exercicis pràctics

L'objectiu de la pràctica és realitzar un programa a Shell Script (bash) que permeti realitzar la preparació de les dades d'un dataset original.

El dataset que utilitzarem en aquesta pràctica és públic i es pot descarregar d'internet en el següent enllaç https://www.kaggle.com/datasets/shivamb/netflix-shows. Addicionalment, per facilitar a l' alumne la seva obtenció, també es pot descarregar del campus virtual. El dataset conté aproximadament 5975 entrades amb pel·lícules i sèries de la plataforma NetFlix. Cada línia del dataset fa referència a una pel·lícula o sèrie, i les columnes contenen la següent informació:

Número	ID Columna	Descripció
1	id	Camp alfanumèric. Conté l'Id del contingut, comença per *tm* si és una pel·lícula i per *ts* si és una sèrie

2	title	Camp alfanumèric. Conté el títol del contingut
3	description	Camp alfanumèric. Conté un text descriptiu del contingut
4	type	Camp alfanumèric. Indica si el contingut és una pel·licula (MOVIE) o una sèrie (SHOW)
5	release_year	Camp numèric. Indica l'any en que es va fer públic el contingut.
6	age_certification	Camp alfanumèric. Indica la classificació del contingut d'acord a la normativa dels Estats Units, podeu trobar el significat de cada cas aquí per pel·lícules i aquí per sèries.
7	runtime	Camp numèric. Durada del contingut en minuts.
8	genres	Llista. Conté la llista de gèneres en que s'ha classificat el contingut, ex. 'documentation', 'comedy', 'action', etc. La llista està delimitada per [] i la separació entre elements és l'espai en blanc.
9	production_countries	Llista. Conté la llista (mateix format que per la columna 9) d'acrònims dels països participants en la producció del contingut.
10	seasons	Camp numèric (real). Conté el número de temporades del contingut (buit si és una pel·licula)
11	imdb_id	Camp alfanumèric. Conté l'identificador del contingut a la base de dades "Internet Movie Database" (més informació sobre aquesta base de dades)
12	imdb_score	Camp numèric (real). Indica la puntuació (sobre 10) IMDb del contingut (la classificació és diferent per pel·lícules i sèries).
13	imdb_votes	Camp numèric. Indica el número de vots que ha rebut aquest contingut per part dels usuaris de la plataforma IMDtmdbb.
14	tmdb_popularity	Camp numèric (real). Indica la popularitat del contingut d'acord a la base de dades "The Movie Database" (TMDB).
15	tmdb_score	Camp numèric (real). Indica la puntuació (sobre 10) TMDB del contingut (la classificació és diferent per pel·lícules i sèries).

Taula 1: Definició de camps del Dataset

A partir d' aquest dataset, es demana realitzar un script utilitzant Bash que automatitzi les següents operacions sobre el conjunt de dades original:

Adaptació de les dades

1. El primer pas de processament que es vol realitzar consisteix a eliminar totes les files que no comencin amb un identificador de contingut vàlid (de la forma tmXXXXX o tsXXXXX). L'script ha d'indicar després d'aquest pas el nombre de registres (línies) eliminats.

2. Després s'eliminaran del dataset els registres on el nom del contingut estigui en un alfabet desconegut. Això vol dir que eliminarem el registre (línia) de qualsevol contingut amb un títol que no comenci per un caràcter alfanumèric (lletres minúscules o majúscules i números) o pels caràcters “, #, ‘, ¿, ¡.
3. Com tenim camps que només tenen sentit per pel·lícules i d’altres que només el tenen per les sèries, el que farem abans de continuar amb la neteja de les dades és separar els dos tipus de continguts. Així que crearem un arxiu Movies.csv i un arxiu Shows.csv on emmagatzemarem les pel·lícules i les sèries respectivament. Comproveu que la suma del número de registres en cada dataset sigui igual al del dataset original.
4. Per l’estudi que ens demanen fer, és necessari que els camps *imdb_score*, *imdb_votes*, *tmdb_popularity* i *tmdb_score* tinguin valors. Així que eliminarem els camps dels arxius Movies.csv i Shows.csv ens els quals alguns d’aquests camps estigui buit. L’script ha d’indicar després d’aquest pas el nombre de registres (línies) eliminats.
5. Ara que ja tenim les dades “netes”, ens demanen afegir informació extra al dataset. Tant per la informació que bé de IMDB com de TMDB tenim una puntuació (score) i un indicador de popularitat (imdb_votes, tmdb_popularity), però no tenim un índex que relacioni aquests dos aspectes. La qüestió és que un sèrie o pel·lícula pot tenir una nota molt alta obtinguda amb pocs vots, la qual cosa pot conduir per exemple a que un algorisme d’aprenentatge automàtic li doni una importància que potser no tindria amb un nombre més alt de vots.
   Si definim un índex que relacioni la nota donada a un contingut amb la seva popularitat relativa, tindrem una informació més fiable sobre la valoració real d’aquest contingut. Així, definim com a coeficient de fiabilitat la següent relació: score x (popularitat/max(popularitat)). En el nostre cas, això es tradueix en imdb_score x (imdb_votes/max(imdb_votes)) per la informació que ve d’IMDB i en tmdb_score x (tmdb_popularity/max(tmdb_popularity)) per la informació que ve de TMDB.
   Aquests càlculs els afegirem en dues noves columnes (la 16 i la 17) que tindran per títols imdb_reliability i tmdb_reliability respectivament.
   Nota: per assolir aquest punt cal fer servir l’ordre awk, ja que amb aquesta ordre es pot treballar amb càlculs amb nombres reals.
6. Per acabar, farem uns quants tests per estar segurs que els datasets obtinguts són correctes, programeu en l’script les ordres necessàries per saber:
   a. Quina pel·lícula té la millor puntuació IMDB? (mostreu només els camps id, title, production_countries i imdb_score)
      Resultat: tm76557,No Longer Kids,['EG'],9.0

   b. El mateix per la sèrie amb millor puntuació IMDB.
      Resultat: ts4,Breaking Bad,['US'],9.5

   c. Quina és la pel·lícula més popular segons IMDB? ? (mostreu només els camps id, title, production_countries i imdb_votes)
      Resultat: tm92641,Inception,['GB' 'US'],2268288.0

   d. El mateix per la sèrie amb més popularitat IMBD.
      Resultat: ts4,Breaking Bad,['US'],9.5

   e. Quina pel·lícula té la millor coeficient de fiabilitat IMDB? (mostreu només els camps id, title, production_countries i imdb_reliability)
      Resultat: tm92641,Inception,['GB' 'US'],8.8

f. El mateix per la sèrie amb millor coeficient de fiabilitat.
Resultat: ts4,Breaking Bad,['US'],9.5

g. Repetiu els 6 apartats anteriors per l'índex TMDB.
Resultats:
tm362264,The Gift,['TZ'],10.0
ts188031,Pink Zone,['US'],10.0
tm1064065,Black Crab,['SE'],944.405
ts20110,Peaky Blinders,['GB'],971.727
tm1082564,The Adam Project,['US'],6.81614
ts20110,Peaky Blinders,['GB'],8.6

Consultes simples

1. Finalment, si l'script rep com a únic ***paràmetre d'entrada una cadena de caràcters***, en lloc de realitzar tot el processament descrit en els passos anteriors (1-6), ha d'imprimir el registre o registres (línies) que continguin aquesta cadena en els dos arxius, en cas de no trobar cap registre, haurà d'imprimir un missatge indicant que no s'han trobat coincidències.

El desenvolupament dels punts anteriors es podrà dur a terme de forma lliure, és a dir, l'alumne podrà utilitzar les ordres que consideri més oportunes en cada moment, podent haver-hi diverses ordres per realitzar una mateixa operació, sempre que siguin ordres estàndard de Linux (incloent-hi l'AWK). Ara bé, atès que el nostre objectiu no és la programació en si mateixa, us donem algunes pistes sobre com organitzar el programa:

1. *Primer comprovem l'existència o no d'arguments, si no hi ha:*
    a. *Còpiem la línia d'encapçalat + els noms dels dos nous camps que inclourem a l'arxiu de resultats*
    b. *Decidim com vam processar l'arxiu d'entrada. Usarem una estructura iterativa o l'ordre awk? En el primer cas tindrem un script més lent i, en el fons, més complex, però és possible que sent inexperts en aquestes tasques us sentiu més còmodes fent servir aquesta solució.*
    c. *Determinem quins són els camps que utilitzem per decidir quins registres estaran a l' arxiu de resultats. Si utilitzem el comandament awk per resoldre aquest cas, haurem de construir el patró corresponent. Si, per contra, utilitzem una estructura iterativa, haurem de fer servir un if amb totes les condicions necessàries.*
    d. *Per a cada registre haurem de calcular els dos camps extres que es demanen. En el cas d'utilitzar una estructura iterativa, és difícil calcular una divisió real, per tant, haureu de fer servir l'awk*
    e. *Finalment, afegim el registre + els dos nous camps a l'arxiu de resultats.*
2. *Si hi ha arguments:*
    a. *N'hi ha prou amb buscar el(s) registre(s) que continguin la cadena passada com a argument (si hi ha més arguments, els ignorem) i mostrar per pantalla el seu contingut o un missatge indicant que no es troba.*

Si heu fet els tutorials, sou capaços de fer cadascun dels passos indicats. Feu-ho de forma incremental, comprovant que obteniu el resultat esperat abans d'anar al següent pas.

Recorda que les hores de classe han de ser complementades amb aproximadament el doble d'hores de treball autònom. Això és especialment cert per a les pràctiques de laboratori. No és possible (en general) acabar les pràctiques només en les hores de laboratori.

Recordat que per poder executar l'script caldrà donar-li permisos d'execució, això ho podem fer executant la següent línia en la termina de Linux:

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

4. AVALUACIÓ

Per a l' avaluació de la pràctica es tindran en compte els següents elements:

1. **Funcionament**. Si no s' aconsegueixen totes les funcionalitats demanades es reduirà la nota final. Aquest serà l'apartat de major pes (45%)
2. **Memòria**. Que ha d'incloure la descripció del treball realitzat (anàlisi del problema i disseny de la solució), així com de les dificultats més importants que us hauré trobat (20%)
3. **Qualitat de la solució**. Es valorarà la qualitat del codi desenvolupat (ús de les ordres, organització de la solució, documentació i presentació) (15%)
4. **Assistència a les sessions de pràctiques i participació en el seguiment realitzat pel professor de pràctiques**. (20%)

En aquest curs hem introduït l'ús del programari de control de versions git i la seva versió al núvol GitHub. Amb l'objectiu d'animar-vos a provar el seu ús, hem decidit que aquells grups que demostrin que han utilitzat aquesta plataforma de forma correcta durant el desenvolupament de tots els exercicis pràctics de l'assignatura seran premiats amb un punt extra en la part pràctica de l'assignatura.

Aquest punt no podrà en cap cas usar-se per aprovar les pràctiques (si no s'arriba al 5, les pràctiques estaran suspeses independentment que s'hagi fet servir correctament git i GitHub).

Perquè el vostre professor de pràctiques pugui valorar si s'ha utilitzat de forma correcta git + GitHub haureu d'incloure en la memòria de cada exercici l'enllaç al repositori de la pràctica a GitHub.