PAC1: Recomanadors i agrupació

Presentació

En aquesta prova d'avaluació estudiarem com aplicar tècniques de recomanació i agrupació partint de la base de dades *MovieLens*-100k.

Competències

En aquest enunciat es treballen en un determinat grau les següents competències general de màster:

- Capacitat per a projectar, calcular i dissenyar productes, processos i instal·lacions en tots els àmbits de l'enginyeria en informàtica.
- Capacitat per al modelat matemàtic, càlcul i simulació en centres tecnològics i d'enginyeria d'empresa, particularment en tasques d'investigació, desenvolupament i innovació en tots els àmbits relacionats amb l'enginyeria en informàtica
- Capacitat per a l'aplicació dels coneixments adquirits i de solucionar problemes en entorns nous o poc coneguts dins de contextes més amplis i multidisciplinars, essent capaços d'integrar aquests coneixements.
- Posseir habilitats per a l'aprenentatge continuat, autodirigit i autònom.
- Capacitat per a modelar, dissenyar, definir l'arquitectura, implantar, gestionar, operar, administrar y mantenir aplicacions, xarxes, sistemes, serveis i continguts informàtics.
- Capacitat per assegurar, gestionar, auditar i certificar la qualitat dels desenvolupaments, processos, sistemes, serveis, aplicacions i productes informàtics.

Les competències específiques d'aquesta assignatura que es treballen són:

- Entendre que és l'aprenentatge automàtic en el context de la Intel·ligència Artificial
- Distingir entre els diferents tipus i mètodes d'aprenentatge
- Aplicar les tècniques estudiades a un cas concret

Objectius

En aquest PAC es practicaran els conceptes del temari relacionat amb recomanació i clustering, en una vertent clarament pràctica enmarcada en l'ús de llibreries Python.

Descripció de la PAC

En aquesta PAC estudiarem en profunditat l'ús de la llibreria Surprise (http://www.surpriselib.com) com a eina en Python que facilita l'implementació de recomanadors basats en diferents tècniques d'agrupació (*clustering*).

Farem servir la coneguda base de dades *MovieLens*-100k (https://grouplens.org/datasets/movielens/).

Important: Com farem un ús intensiu d'aquesta llibreria, aquesta PAC requereix implícitament un estudi i consulta actius de l'apartat "Documentation" de la pàgina web anteriorment esmentada.

Per a aquesta PAC, se us facilita un fitxer zip amb dos fitxers Python. Aquests dos fitxers hauran de ser completats per l'alumen omplint els diversos forats que s'hi indiquen amb el codi que correspongui.

NO MODIFIQUEU cap nom de funció, variable, ni tampoc dels propis fitxers. La PAC s'ha de completar tot fent servir la base que es proporciona, que en moltes ocasions pot guiar-vos en la manera com dur a terme els exercicis proposats.

Com podeu veure, al fitxer *pac1.py* tenim diverses definicions de funcions incompletes i un programa molt simple que serà executat (al final del fitxer). Al fitxer *IAA_predictors.py* tenim, de manera anàloga, la definició d'unes classes sobre les que treballarem.

Exercici 1 (2 punts)

a) (1 punt) Completeu la definició del mètode *predict*. Aquest mètode rep un *dataset* prèviament carregat, un identificador d'usuari i un identificador d'ítem, i la seva finalitat és preparar una sèrie d'algorismes de predicció (basats en diferents tècniques d'agrupació) per tal de dur a terme una predicció de la puntuació (*rating*) amb què

l'usuari *uid* valorarà l'ítem *iid*. Com podeu veure al codi, aquest mètode realitzarà aquesta predicció partint dels algorismes *SVD, KNNBasic, KNNWithMeans* i *NormalPredictor*. Un cop definit el mètode, dueu a terme les prediccions que aquests algorismes faran de la valoració de l'usuari 777 de l'ítem 333.

b) (1 punt) Consulteu la documentació i expliqueu com proporcionen la seva predicció els algorismes *KNNWithMeans* i *NormalPredictor.*

Exercici 2 (2 punts)

a) (1 punt) Completeu la definició del mètode *multiple_cv*. Aquest mètode rep un *dataset* prèviament carregat, i el nombre de vies (o *folds*, per defecte 5) per tal de dur a terme una valoració creuada dels mateixos algorismes fets servir a l'Exercici 1. Expliqueu les característiques i els avantatges de la validació creuada i justifiqueu per què és important considerar diverses particions del *trainset* per validar la bondat del nostre sistema.

b) (1 punt) Identifiqueu el mètode que dona pitjors resultats i raoneu per què creieu que aquest mètode no és adient. En quines condicions seria més raonable utilitzar-ho com a predictor?

Exercici 3 (2 punts)

Com ja heu observat, les valoracions registrades a *MovieLens* corresponen a valors discrets en l'interval [1,5], segurament obtinguts després que l'usuari valorés una determinada pel·lícula amb una quantitat d'estrelles.

Imaginem que treballem com a enginyers al portal web d'on s'han extret aquestes puntuacions, i des del departament d'*user experience* ens comuniquen que han realitzat un estudi que suggereix que hem de canviar aquesta manera de valorar les pel·lícules, de manera que ara, en comptes dels valors de l'1 al 5, seran valorades escollint una de les següents opcions:

```
["Molt dolenta", "Pobre", "No està malament",
"Entretinguda", "Recomanable", "Molt bona", "Brillant"]
```

Quines consideracions hem de tenir en compte per a poder redissenyar el sistema de recomanació? Ens servirà el conjunt de dades d'entrenament que ja teníem? En cas d'haver d'introduïr modificacions a la implementació del

sistema, indiqueu-les, tot justificant cadascuna de les decisions que prendríeu.

Quines implicacions tindria el fet que al cap de 2 mesos ens indiquessin que hem d'afegir l'opció "Decepcionant" entre "Pobre" i "No està malament"?

Important: per realitzar aquest exercici no cal implementar cap programa, només un raonament justificat i complet.

Exercici 4 (4 punts)

Estudieu detingudament com crear un algorisme de predicció propi (http://surprise.readthedocs.io/en/stable/building_custom_algo.html) i preneu com a exemple la definició dels algorismes que la llibreria Surprise porta implementats per defecte (per exemple, https://github.com/NicolasHug/Surprise/blob/master/surprise/prediction_algorithms/knns.py).

En aquest exercici definirem un parell d'algorismes propis. Consulteu el fitxer IAA_predictors.py, on hi ha definida una classe arrel *SymmetricAlgo*, de la qual heretaran els nostres algorismes, i a continuació tenim definicions incompletes de dos algorismes. En aquest context es demana:

a) (1.5 punts) Completar la definició de la classe *Dummy_IAA*, de manera que el mètode *estimate* simplement retorni una valoració a l'atzar d'entre totes les valoracions fetes per l'usuari que es consulta.

b) (1.5 punts) Completar la definició de la classe *KNN_IAA*, de manera que a la seva inicialització permeti rebre un paràmetre **n**, i el mètode *estimate* faci pràcticament el mateix que el que fa *KNNBasic* de Surprise, però amb la diferència que dels *k_neighbors* triem **n** a l'atzar per calcular el promig ponderat (*weighted average*). En cas que *k_neighbors* tingui una mida inferior a **n**, agafarem el valor de la mida de k_neighbors com a **n**.

c) (1 punt) Completeu el mètode *run_IAAPredictors* de *pac1.py*, i executeu-ho. Aquest mètode es comportarà de dues maneres diferents, segons el paràmetre *do_cv*:

   - Si és True, realitzarà una validació creuada amb 5 vies (*folds*).
   - Si és False, utilitzarà els paràmetres *uid* i *iid* per fer una predicció de valoració de l'ítem *iid* per part de l'usuari *uid*.

   En ambdós casos, com podeu veure al codi, es faran servir els algorismes Dummy_IAA i KNN_IAA, un darrere l'altre.

Recursos

Aquesta PAC requereix els recursos següents:

Bàsics:

Per a realitzar aquesta PAC disposeu d'uns fitxers adjunts:

- *pac1.py*
- *IAA_predictors.py*

Complementaris: manual de teoria de l'assignatura, vídeos de l'assignatura, web de Surprise lib.

Criteris de valoració

La valoració de cada exercici està indicada en l'enunciat de cadascun d'ells.

S'ha d'incloure el codi font de les vostres solucions a l'entrega.

A més, s'ha de respondre cada exercici d'aquest raonant les accions preses al codi i el perquè, i les reflexions que n'extraieu del resultat de les operacions.

Format i data de lliurament

La PAC s'ha de lliurar abans del **proper 25 de març** (inclòs).

La solució ha de consistir en un arxiu zip que contingui un informe en format pdf i els arxius en format python (*.py) que corresponguin a la solució adoptada.

Adjunteu l'arxiu a un missatge en el apartat de **Lliurament i Registre de AC (RAC)**. El nom de l'arxiu ha de ser CognomsNom_IAA_PAC2 amb extensió zip. NO CANVIEU EL NOM DEL FITXER DEL CODI FONT BASE, ni el nom de cap declaració de variable o funció.

Per a dubtes i aclaracions sobre l'enunciat, dirigiu-vos al consultor responsable de l'aula.

Nota: **Propietat intel·lectual**

Sovint és inevitable, en produir una obra multimèdia, fer ús de recursos creats per terceres persones. És per tant comprensible fer-ho en el marc d'una pràctica dels estudis del Grau Multimèdia, sempre i això es documenti clarament i no suposi plagi en la pràctica.

Per tant, en presentar una pràctica que faci ús de recursos aliens, s'ha de presentar juntament amb ella un document en què es detallin tots ells, especificant el nom de cada recurs, el seu autor, el lloc on es va obtenir i el seu estatus legal: si l'obra està protegida pel copyright o s'acull a alguna altra llicència d'ús (Creative Commons, llicència GNU,

GPL ...). L'estudiant haurà d'assegurar-se que la llicència que sigui no impedeix específicament seu ús en el marc de la pràctica. En cas de no trobar la informació corresponent haurà d'assumir que l'obra està protegida pel copyright.

Hauran, a més, adjuntar els fitxers originals quan les obres utilitzades siguin digitals, i el seu codi font si correspon.
Un altre punt a considerar és que qualsevol pràctica que faci ús de recursos protegits pel copyright no podrà en cap cas publicar-se en Mosaic, la revista del Graduat en Multimèdia a la UOC, a no ser que els propietaris dels drets intel·lectuals donin la seva autorització explícita.