22.403 - Programació per a la ciència de dades
PAC 4

En aquest document trobareu l'enunciat de la quarta activitat d'avaluació continuada (PAC) de l'assignatura. Aquesta PAC constarà d'un únic exercici a resoldre, que engloba molts dels conceptes coberts durant l'assignatura.

El context

L'empresa UOC Sound Dynamics us ha contractat com a científics de dades per un projecte que té com objectiu crear una aplicació per descobrir artistes musicals. De moment el projecte està en una fase inicial on s'han d'analitzar les dades existents i així ajudar a definir el producte final que més tard implementaran els desenvolupadors. La vostra feina és crear un paquet de Python que llegeixi les dades, les prepari per l'anàlisi, en calculi algunes estadístiques bàsiques, i creï visualitzacions per comparar diferents artistes.

Les dades que s'utilitzaran en l'anàlisi contenen informació sobre cançons, sobre els àlbums que les contenen, i sobre els artistes que les han creat. A part de detalls bàsics com per exemple, el nom dels artistes o el títol de les cançons, les dades més interessants que seran la base de l'anàlisi són les audio features que ens proporciona Spotify. Podeu veure més detall de les dades proporcionades a la secció *Dades*.

Objectius del projecte

El Data Lead del projecte ja ha avançat la feina d'especificació i ha dissenyat una primera fase que comença el **17/12/2021** i acaba el **10/01/2022**. En aquesta fase s'han de fer les tasques que es descriuen a la secció *Tasques*. Cada tasca conté una descripció i un llistat de criteris d'acceptació que s'han de complir per donar la tasca per finalitzada. A més a més, s'han definit alguns aspectes generals a tenir en compte:

- Aquest paquet serà utilitzat en futures fases del projecte per tant la implementació s'ha de fer en arxius plans de Python i no en Jupyter Notebooks.
- S'ha d'implementar el codi en funcions que siguin el més generals possibles, sempre que sigui possible i que tingui sentit, per així poder-les utilitzar en fases posteriors del projecte.
- El paquet ha d'incloure el corresponent codi organitzat lògicament (separat en mòduls i organitzats per funcionalitat), la documentació de codi (docstrings) i tests. A més, s'hauran d'incloure els corresponents arxius de documentació d'alt nivell (README), així com els arxius de llicència i dependències (requirements.txt). Fer un setup.py és opcional, però si es fa es valorarà positivament.

Les dades

Les dades de cançons, àlbums i artistes es troben en 3 CSVs diferents dins de l'arxiu comprimit *data.zip*. Fixeu-vos que les dades estan normalitzades i al llistat de *tracks* (cançons) hi ha els identificadors de l'artista (*artist_id*) i de l'album (*album_id*) per poder relacionar-los.

tracks_norm.csv

artist_id: identificador de l'artista **album_id**: identificador de l'àlbum **track_id**: identificador de la cançó **track_sp_id**: identificador de la cançó a Spotify
name: títol de la cançó **number**: número de cançó dins de l'àlbum **disc_number**: número d'àlbum (pel cas de àlbums dobles) **popularity**: indicador de popularitat **preview_url**: URL per poder escoltar un tros de la cançó **duration_ms**: duració en ms
audio features de la cançó: danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, tempo, time_signature

albums_norm.csv

artist_id: identificador de l'artista **album_id**: identificador de l'àlbum **album_sp_id**: identificador de l'àlbum a Spotify
name: títol de l'àlbum **popularity**: indicador de popularitat **release_year**: any de publicació **total_tracks**: número de cançons a l'àlbum

artists_norm.csv

artist_id: identificador de l'artista **artist_sp_id**: identificador de l'artista a Spotify
name: nom de l'artista **popularity**: indicador de popularitat **followers**: número de seguidors a Spotify **total_albums**: número d'àlbums

Tasques

Tasca 1: Crear dataset de tracks des-normalitzat

Processeu els tres arxius originals continguts a l'arxiu comprimit *dades.zip* per tal de crear un dataset final de *tracks* que inclogui també les dades dels àlbums i artistes corresponents.

S'ha de tenir en compte que degut a errors d'extracció de les dades des dels sistemes origen:

- Alguns noms d'artistes no tenen el format correcte. Aquest ha de tenir cada paraula del nom en majúscula (per exemple 'the beatles' hauria de d'estar registrat com 'The Beatles')
- Alguns registres de popularity del dataset de tracks no tenen cap valor. Per aquest anàlisi assumirem que les tracks sense *popularity* tindran el valor mig del de totes les tracks.

Altres aspectes a tenir en compte:

- L'arxiu amb els datasets originals és un zip comprimit i s'haurà de descomprimir programàticament.
- Es recomana utilitzar mètodes de la llibreria pandas per llegir i processar els arxius anteriors.

Criteris d'acceptació:

- Mostrar per pantalla un comentari amb el número de *tracks* total i el nombre de columnes del dataset final.
- Mostrar per pantalla un comentari amb quantes tracks no tenen valor de *popularity*.

Tasca 2: Explorar alternatives de lectura de fitxers

Exploreu mètodes alternatius per llegir columnes d'un arxiu CSV. Tot i que els arxius actuals tenen una mida reduïda, volem observar el comportament quan la mida d'aquests arxius sobrepassi 1Gb.

Per aquest estudi volem trobar una manera més eficient de llegir una columna comparat amb fer la mateixa acció utilitzant la llibreria pandas. Per fer-ho s'han d'implementar dues funcions amb la següent nomenclatura:

- *get_column_pandas*: implementarà la lectura mitjançant pandas.
- *get_column_{your_method}*: implementarà la lectura mitjançant el mètode escollit per vosaltres. Heu de substituir *{your_method}* per un nom que identifiqui d'alguna manera l'algortime escollit.

Les dues funcions rebran com a paràmetres d'entrada:

- la ruta de l'arxiu CSV a llegir.
- el nom de la columna de l'arxiu CSV que volem llegir.

Les dues funcions hauran de retornar una llista amb tots els valors de la columna.

Es vol comparar els temps d'execució de cada una de les versions variant els paràmetres d'entrada segons el que s'especifica als criteris d'acceptació. Per poder avaluar el temps d'execució d'arxius de diferent mida, es proposa utilitzar els 3 arxius originals que s'han utilitzat a la tasca 1 donat que són de mides diferents (el d'artistes essent el més petit, i el de cançons essent el més gran). Realitzeu execucions per les dues versions utilitzant els 3 datasets originals disponibles llegint en cada cas les següents columnes:

- artists_norm.csv: *artist_id*
- albums_norm.csv: *album_id*
- tracks_norm.csv: *track_id*

Per facilitar l'anàlisi i la comparació, es demana crear un gràfic que compari les dues versions de la funció de lectura amb el número de files llegides a l'eix horitzontal i el temps emprat a l'eix vertical.

Criteris d'acceptació:

- Mostrar per pantalla el gràfic que mostra la comparació entre les dues versions de la funció.

Tasca 3: Filtratge i comptadors bàsics

Realitzeu un anàlisi exploratori inicial de les dades utilitzant el dataset de tracks creat a la Tasca 1 i responent:

A. Quantes tracks hi ha de l'artista *Radiohead*?
B. Quantes tracks contenen la paraula *'police'* al títol?
C. Quantes tracks són d'àlbums publicats a la dècada dels 1990?
D. Quina és la track amb més popularitat dels últims 10 anys?
E. Quins artistes tenen tracks a cada una de les dècades des del 1960?

S'ha de remarcar que l'objectiu és implementar les funcions necessàries per respondre les preguntes anteriors però que a l'hora puguin ser reutilitzades en futures fases del projecte i que en facin el manteniment senzill.

Criteris d'acceptació:

- Mostrar per pantalla un comentari per cada una de les preguntes anteriors

Tasca 4: Anàlisi inicial d'*audio features*

En aquesta tasca volem endinsar-nos en l'anàlisi de les dades d'audio features per començar a entendre-les i a interpretar-les.

Heu de:

A. Calcular el mínim, la mitjana i el màxim de la feature *energy* de totes les tracks de *Metallica*.
B. Calcular la mitjana de la feature *danceability* de cada àlbum de *Coldplay* i crear una gràfica per visualitzar el resultat.

Criteris d'acceptació:

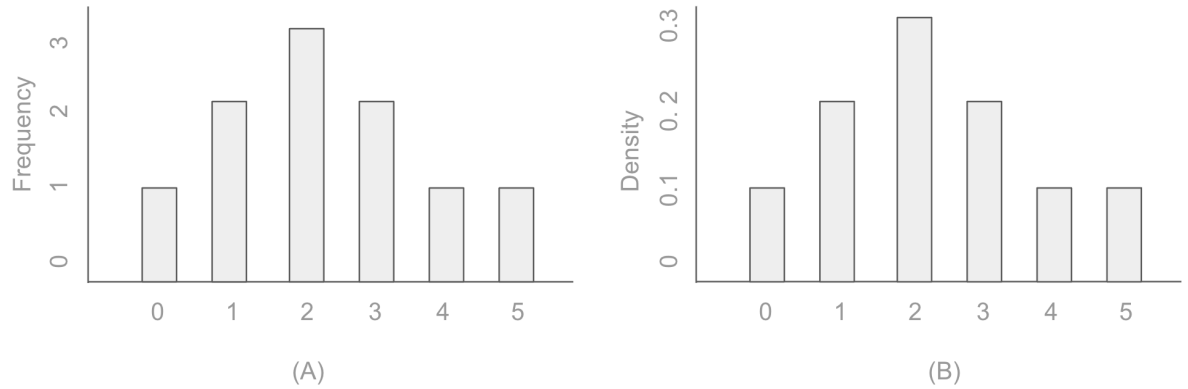
- Mostrar per pantalla un comentari amb les estadístiques bàsiques de A.
- Mostrar per pantalla la gràfica generada a B.

Tasca 5: Histograma d'una *audio feature* d'un artista

A part de l'anàlisi estadística bàsica realitzada a la tasca anterior, es necessita una forma visual que mostri les dades d'una audio feature de l'artista en qüestió. S'ha decidit explorar la visualització en format histograma on a l'eix horizontal conté els diferents valors que pren la audio feature i a l'eix vertical hi ha la densitat de probabilitat de cada valor. És important entendre que la densitat de probabilitat no és la freqüència (el número de vegades que es dóna el valor en concret) sinó una normalització de la freqüència.

A mode d'exemple, a la figura a continuació es mostren dos histogrames de les dades contingudes a la llista [0, 1, 1, 2, 2, 2, 3, 3, 4, 5] de N=10 elements.

- (A) histograma de freqüència: quantes vegades apareix cada número.
- (B) histograma de densitat de probabilitat: quantes vegades apareix cada número dividit pel número total de mostres.



Exemple d'histograma de (A) freqüència i (B) de densitat de probabilitat. Dades de N=10 mostres

Per més detall podeu consultar aquí.

L'objectiu de la tasca és doncs crear una funció que rebi les dades i configuració necessària per crear l'histograma i visualitzar-lo.

Criteris d'acceptació:

- Mostrar per pantalla un histograma que mostri la *acousticness* de les cançons de *Ed Sheeran*.

Tasca 6: Comparar artistes visualment

Una vegada tenim l'histograma d'una sola *audio feature* i d'un un sol artista, la part interessant és utilitzar aquest tipus de visualització per comparar dos artistes.

S'ha de crear una versió de l'histograma anterior que inclogui les dades de dos artistes superposant els dos histogrames un sobre l'altre. És important buscar una forma de visualitzar els dos histogrames superposats sense que es tapin un a l'altre.

Com que estem comparant dos artistes, aquí cobra molta importància utilitzar un histograma que visualitzi la densitat de probabilitat i no la freqüència perquè si comparem dos artistes amb un número molt diferent de *tracks* la visualització no ajudarà a analitzar-los perquè les alçades dels histogrames serien molt diferents. En canvi, visualitzant la densitat de probabilitat eliminem l'efecte d'aquesta diferència.

Com en las tasques anteriors, és important fer una implementació utilitzant funcions parametritzades que puguin ser reutilitzades de forma fàcil.

Criteris d'acceptació:

- Mostrar per pantalla un histograma que compari la *energy* de *Adele* i la de *Extremoduro*.

Tasca 7: Calcular similitud entre artistes

Una altra forma de comparar dos artistes és fer-ho utilitzant un càlcul matemàtic. Com que per cada track tenim 12 audio features, podem obtenir la mitjana de les audio features per cada artista (un vector de 12 valors) i després comparar dos artistes comparant els seus dos vectors d'audio features.

S'ha d'explorar la millor forma de comparar artistes i per això s'analitzaran dues formes de calcular la similitud entre ells:

- Similitud euclidiana
- Similitiud cosinus

Per cada una de les mètriques s'ha de construir una matriu amb els valors de similitud entre tots els parells d'artistes. És a dir, a la posició (1,3) hi ha la similitud entre l'artista 1 i l'artista 3. Una vegada calculada la matriu s'ha de visualitzar en una imatge tipus *heatmap* per poder veure ràpidament la similitud entre ells.

És important tenir en compte que la implementació per crear la matriu ha d'estar pensada pel futur i ha de ser senzill poder utilitzar una nova mètrica.

Criteris d'acceptació:

- Mostrar per pantalla els *heatmap* de la similitud euclidiana i de la similitud cosinus comparant els artistes *Metallica*, *Extremoduro*, *AC/DC* i *Hans Zimmer*.

Tasca 8 (Opcional): Crides a API externa

Per completar el dataset original d'artistes ens interessa algunes dades que es poden obtenir a través de la API de AudioDB: any de formació de l'artista i ciutat d'origen.

L'objectiu de la tasca és crear un dataset independent amb les columnes *artist_name*, *formed_year*, *country*. Aquest dataset s'ha de guardar en un arxiu csv anomenat *artists_audiodb.csv*.

S'ha de tenir en compte que aquesta implementació ha de ser eficient en el cas que s'hagi de buscar molts artistes, cosa que implicaria moltes crides a la API remota i, per tant, molts instants d'espera fins a obtenir la resposta.

Criteris d'acceptació:

- Mostrar per pantalla les dades obtingudes (*artist_name*, *creation_year*, *country*) dels artistes *Radiohead*, *David Bowie* i *Måneskin.*
- Avalueu la vostra implementació descarregant la informació de tots els artistes als datasets originals i comenteu perquè creieu que la vostra implementació és eficient i com creieu que escalarà si s'han de buscar milers d'artistes simultàniament.

Altres consideracions

Tests

Es demana també que el codi estigui degudament documentat i testejat, assegurant una cobertura del codi font de com a mínim un 50%. Es recomana l'ús de la llibreria unittest degut el seu ús en anteriors projectes.

El mesurament de la cobertura dels testos s'utilitza per avaluar l'eficàcia dels testos proposats. En particular, serveix per determinar la qualitat dels testos desenvolupats i per determinar les parts crítiques del codi que no han estat testejades. Per tal de mesurar aquest valor, proposem l'ús de l'eina Coverage.py. A la documentació, podreu trobar com instal·lar-la i com usar-la.

Guia d'estil

Es demana també que el codi segueixi la guia d'estil de PEP8. Per fer-ho es recomana l'ús de l'eina black ja que ens permetrà automatizar el formateig del nostre codi segons la guia d'estil de Python. Podeu trobar més detalls sobre la instal·lació i configuració d'aquesta eina en el següent enllaç.

Ús de Git

Per tal de posar en pràctica el que heu après a la Unitat 6 sobre Git, proposem l'ús de GitHub Classroom per a desenvolupar el vostre paquet de Python. GitHub Classroom és una eina gratuïta de codi obert que ajuda a simplificar l'ús educatiu de GitHub. Hem usat GitHub Classroom per a crear una aula com aquesta i on hem creat una tasca per a la PAC4. Per a fer ús d'aquest espai que hem creat, us aconsellem seguir els passos indicats en aquesta guia que expliquen com crear un repositori per a treballar en la tasca que hem preparat i trobareu en aquest enllaç.

L'ús d'aquesta eina no és obligatori per l'avaluació de la PAC4, però creiem que és una molt bona oportunitat per a posar en pràctica els vostres coneixements en un entorn vital per a tothom que treballi o vulgui treballar en l'àmbit de la ciència de dades.

Criteris de correcció

Aquesta PAC es valorarà seguint els criteris següents:

- **Funcionalitat** (6 punts): Es valorarà que el codi implementi correctament el que demana l'enunciat.
    - Tasca 1 (1 punt)
    - Tasca 2 (0.75 punts)
    - Tasca 3 (0.5 punts)
    - Tasca 4 (0.5 punts)
    - Tasca 5 (0.75 punt)
    - Tasca 6 (0.75 punt)
    - Tasca 7 (1 punt)
    - Visualitzacions (0.75 punt)
    - Tasca 8 (1 punt extra)
- **Documentació** (0.5 punts): Totes les funcions dels exercicis d'aquesta PAC hauran d'estar correctament documentades utilitzant docstrings (en el format que preferiu).
- **Modularitat** (0.5 punt): Es valorarà la modularitat del codi (tant l'organització del codi en fitxers com la creació de funcions).
- **Estil** (0.5 punts): El codi ha de seguir la guia d'estil de Python (PEP8), exceptuant els casos on fer-ho compliqui la llegibilitat del codi.
- **Tests** (1.5 punts): El codi ha de contenir una o diverses *suites* de testos que permetin comprovar el bon funcionament de les funcions implementades, obtenint un mínim del 50% de cobertura.
- **Requeriments** (0.5 punts): Hi haurà d'haver un fitxer de requeriments que llisti (només) les llibreries necessàries per a executar el codi.
- **README** i **llicència** (0.5 punts): Es valorarà la creació d'un fitxer de README, que presenti el projecte i expliqui com executar-lo, així com la inclusió de la llicència sota la qual es distribueix el codi (podeu triar la que vulgueu).

Important

Nota 1: De la mateixa manera que en les PACs anteriors, els criteris transversals es valoraran de manera proporcional a la part de la funcionalitat implementada.

Per exemple, si el codi només implementa la meitat de la funcionalitat demanada, i la documentació d'aquesta part és perfecta, aleshores la puntuació corresponent a la part de documentació seria de 0.25.

Nota 2: És imprescindible que el fitxer de README que inclogueu expliqui clarament com s'ha d'executar el vostre codi per tal de generar les gràfiques resultants de l'anàlisi i tots els resultats, a més de com executar els tests i comprovar la cobertura.

Nota 3: Lliureu el paquet com a un únic arxiu .zip al Registre d'Avaluació Continua. **El codi Python haurà d'estar escrit en fitxers plans de Python.**