

Langage de programmation 1 : Projet

Comme dit tout au long de ce cours, pratiquer est la meilleure façon d'apprendre efficacement à programmer en Python. Vous aurez l'occasion, à travers ce projet, de valider vos connaissances acquises durant ce module de programmation Python, mais aussi d'approfondir les notions vues et découvrir de nouvelles pratiques.

Consignes générales

- Ce projet sera à réaliser par groupe de **4 élèves**
- Vous devez choisir un sujet parmi les deux proposés. La note finale de ce projet sera calculée sur la base du livrable mais aussi de la qualité du passage à l'oral et des réponses aux questions.
- Votre rendu devra être mis dans un **dossier zippé** nommé :


`sujet_<1 ou 2>_NOMELEVE1_NOMELEVE2_NOMELEVE3_NOMELEVE4.zip`

- Le livrable devra être composé des éléments suivants:
 1. Les données sources utilisées (rassemblées dans un dossier)
 2. Vos scripts Python (regroupés dans un dossier)
 3. Un rapport (au format .pdf) structuré qui doit illustrer votre réflexion pour la réalisation du projet. Il doit comporter à minima les éléments suivants:
 - Le sujet choisi
 - Les analyses demandées en première partie des sujets avec interprétations
 - La démarche de construction de votre programme (+ schéma montrant sa structure, comment les scripts interagissent entre eux)
 - La démarche de construction de l'interface (avec des captures d'écran du rendu visuel)
 - Les instructions pour exécuter votre programme
 - Quelles ont été les difficultés rencontrées ? Comment avez-vous pu (ou pas) les surmonter (au niveau du groupe mais également au niveau individuel)
 - Qu'avez vous appris via ce projet ? (Pour cette partie, une réponse de chaque membre du groupe est attendue)
- L'ensemble du livrable devra être envoyé à mon adresse e-mail (imane.loukah@univ-paris1.fr) pour le dimanche 19 novembre 23:59 dernier délai. **Tout retard sera sanctionné.**
- La soutenance devra reprendre les points principaux de votre réalisation (structure du programme que vous avez créé, principales fonctions, démonstration de son fonctionnement) et décrire brièvement comment le travail a été réparti au sein du groupe. Vous devrez préparer une présentation PowerPoint pour un passage à l'oral de 10 minutes. Les oraux seront réalisés sur la dernière séance du cours (le 21 novembre 2023).

L'utilisation d'IA Générative (type ChatGPT) n'est pas autorisée.

Toute tentative de plagiat sera sanctionnée par la note de 0.

Remarques :

- La qualité et la clarté de vos codes seront pris en considération dans la note du projet. Il sera important de bien organiser vos codes sous forme de fonctions, de bien commenter vos scripts pour que je puisse comprendre votre démarche en les lisant.
-  Vous devrez prendre en compte dans votre code le fait qu'il sera exécuté sur mon pc. Je ne dois pas avoir à modifier à la main tous les chemins que vous écrivez dans vos scripts.
- Attention à la casse pour les différents sujets (plus spécifiquement pour les parties 2 & 3) : si un utilisateur écrit "TOTO", est ce qu'il arrivera à récupérer le résultat "Toto" dans les bases de données ?
- Il n'est pas demandé d'ajouter les résultats de l'analyse de la partie 1 dans l'interface graphique à construire en partie 3
- Toute prise d'initiative supplémentaire sera prise en compte et valorisée

Sujet 1 : Analyse de données Spotify

L'objectif de ce sujet est d'exploiter des données extraites de la plateforme de streaming Spotify. Les données à votre disposition proviennent de Kaggle et vous pouvez directement les télécharger via le lien suivant:

- Source 1 : <https://www.kaggle.com/datasets/yamaerenay/spotify-dataset-19212020-600k-tracks>
- Source 2 : https://www.kaggle.com/datasets/sadeghhoushyar/top-200-most-streamed-songs-on-spotify-2020?select=spotify_top200_global.csv

Il y a dans ces répertoires 3 fichiers de données à utiliser :

- **artists.csv** : Contient des informations sur les artistes

Libellé de la variable	Définition
id	ID de l'artiste
followers	Nombre d'abonnés de l'artiste
genres	Genres associés à l'artiste
name	Nom de l'artiste
popularity	Popularité de l'artiste (entre 0 et 100)

- **tracks.csv** : Contient des informations sur des chansons présentes sur Spotify (chansons datant de 1921 à 2020) - toutes les variables ne sont pas utilisées

Libellé de la variable	Définition
id	ID de la chanson
name	Nom de la chanson
popularity	Popularité de la chanson (entre 0 et 100)
duration_ms	Durée de la chanson en millisecondes
explicit	Si la chanson contient du contenu explicite (1) ou non (0)
artists	Artistes qui figurent sur la chanson
id_artists	ID des artistes qui figurent sur la chanson
release_date	Date de sortie de la chanson
danceability	Echelle qui permet de mesurer si la chanson est propice à la danse ou non (entre 0 et 100).
energy	Echelle qui permet de mesurer l'énergie de la chanson (entre 0 et 100)
key	Note principale de la chanson
loudness	Volume global d'une chanson (en dB)
mode	Modalité (majeure ou mineure) d'une piste, le type de gamme dont est dérivé son contenu mélodique. La majeure est indiqué par la valeur 1 et la mineure par la valeur 0
speechiness	Représente le degré de présence de paroles dans une chanson. Une valeur élevée indique que la chanson est plus vocale, tandis qu'une valeur faible indique une chanson plus instrumentale.
instrumentalness	Mesure entre 0 et 1 pour indiquer si la piste est acoustique (mesure de confiance)
liveness	Détecte la présence d'un public dans l'enregistrement. Plus la valeur est élevée, plus la probabilité que la piste musicale ait été jouée en live.
valence	Echelle qui permet de décrire la positivité musicale véhiculée par une piste. Les pistes avec une valence élevée semblent plus joyeuses tandis que celles avec une valence plus faible semblent plus tristes (entre 0 et 1)
tempo	Tempo global de la chanson (en BPM)
time_signature	Représente la signature rythmique d'une chanson. Indique le nombre de battements par mesure et aide à mesurer la structure rythmique de la chanson

- **spotify_top200_global.csv** : Contient les titres qui font partie du top 200 mondial de l'année 2020

Libellé de la variable	Définition
Artist	Nom de l'artiste
Country	Pays (ici ce sera global comme le fichier est sur le classement mondial)

Libellé de la variable	Définition
Date	Date
Rank	Rang dans le classement
Streams	Nombre d'écoutes
Title	Nom de la chanson

Partie 1 : Analyse descriptive des bases et visualisation:

Note: Cette partie peut être réalisée dans un notebook

Vous devrez tout d'abord effectuer une description classique des données (ex. nombre d'observations, de variables, type des variables...) puis répondre aux questions suivantes:

1. Quels sont les 10 artistes les plus populaires ? Afficher graphiquement leur nombre d'abonnés par ordre décroissant.
2. Calculer le nombre de chansons sorties chaque année. Représenter graphiquement les résultats
3. Quels artistes ont le plus de chansons distinctes dans le top 200 Global ? En cas d'égalité, les ordonner par nombre de streams cumulés décroissants. Représenter graphiquement les résultats
4. Existe-t-il un lien entre la popularité d'une chanson et les autres critères présents dans les données ? (Pour cette question ne pas considérer les variables de la table `spotify_top200_global.csv`)

Partie 2 : Recherche de contenu

Vous devrez ensuite construire un programme qui permettra de retourner à un utilisateur à minima les résultats suivants (vous pouvez rajouter d'autres fonctionnalités si vous le souhaitez) :

- L'utilisateur saisit un nom d'artiste : le programme doit retourner son nombre d'abonnés, les 3 chansons les plus populaires, les 3 chansons les plus récentes ainsi que le nombre de chansons qu'il a dans le top 200 global de 2020 (s'il y en a)
- L'utilisateur saisit un titre de chanson : le programme doit retourner les résultats qui correspondent. Les ordonner par popularité décroissante. S'il y a beaucoup de résultats, n'afficher que les 20 premiers.
- L'utilisateur saisit une année et un genre : le programme doit retourner les chansons correspondant ordonnées par popularité d'artiste décroissante. S'il y a plusieurs chansons pour un même artiste, les ordonner par popularité décroissante

Partie 3 : Création d'interface graphique

Une fois le programme de la partie 2 réalisé, l'idée est de ne pas devoir écrire des lignes de code pour exécuter votre programme une fois le script exécuté. Il faudra donc créer une interface afin que l'utilisateur soit guidé pour l'utilisation de votre programme.

Pour ce faire, vous devrez utiliser la librairie Tkinter¹ pour réaliser une interface graphique. Cette interface devra comporter les éléments suivants:

- Possibilité pour l'utilisateur de saisir les différents critères (selon ce qui sera fait à la partie 2)
- Affichage des résultats selon ce qui a été demandé dans la partie 2 (Idée: insérer des liens vers les pages Wikipedia des artistes quand c'est possible)

*Note: en cas de difficultés, il est possible de faire une version plus « simple » de l'interface en demandant les infos à l'utilisateur avec des **input** puis en affichant les résultats dans la console avec **print**. La version avec Tkinter étant plus avancée, elle sera bien sûr plus valorisée au niveau du barème que la version simple de l'interface.*

Sujet 2 : Analyse de données Amazon US

L'objectif de ce sujet est d'exploiter des données extraites du site de e-commerce Amazon US. Les données à votre disposition proviennent de Kaggle et vous pouvez directement les télécharger via le lien suivant:

- Source : <https://www.kaggle.com/datasets/asaniczka/amazon-products-dataset-2023-1-4m-products>

Il y a dans ce repertoire 2 fichiers de données à utiliser :

amazon_products.csv

Libellé de la variable	Définition
asin	Identifiant Amazon du produit
title	Libellé du produit
imgUrl	URL de l'image du produit
productURL	URL du produit sur Amazon
stars	Note du produit. Si la valeur est égale à 0, cela signifie qu'aucune note n'a été trouvée
reviews	Nombre d'avis (commentaires). Si la valeur est égale à 0, cela signifie qu'aucun avis n'a été trouvé
price	Prix du produit en USD. Si la valeur est égale à 0, cela signifie que le prix n'était pas disponible
listPrice	Prix original du produit, avant réduction. Si la valeur est égale à 0, cela signifie qu'aucune réduction n'est en cours sur le produit

¹ Quelques références pour Tkinter :

- <https://python.doctor/page-tkinter-interface-graphique-python-tutoriel>
- <https://realpython.com/python-gui-tkinter/>
- <https://www.geeksforgeeks.org/python-gui-tkinter/>

Libellé de la variable	Définition
category_id	Identifiant de la catégorie du produit
isBestSeller	True si le produit a eu le label « BestSeller » False sinon
boughtInLastMonth	Nombre de produits vendus le mois dernier, selon Amazon

amazon_categories.csv

Libellé de la variable	Définition
id	Identifiant de la catégorie du produit
category_name	Libellé de la catégorie du produit

Partie 1 : Analyse descriptive des bases et visualisation:

Note: Cette partie peut être réalisée dans un notebook

Vous devrez tout d'abord effectuer une description classique des données (ex. nombre d'observations, de variables, type des variables...) puis répondre aux questions suivantes :

1. Quels sont les 10 articles ayant le plus d'avis ? Afficher graphiquement leur note moyenne respective par ordre décroissant.
2. Calculer le prix moyen des produits par catégorie. Afficher graphiquement les prix moyens pour les 15 catégories les plus chères
3. Quels sont les produits Best Seller ayant une note inférieure à 4/5 ? Afficher la distribution des prix pour ces produits. Quels sont les 5 qui ont été le plus vendus ? Combien d'unités ?
4. Existe-t-il un lien entre la note d'un produit et les autres critères présents dans les données ?

Partie 2 : Recherche de contenu

Vous devrez ensuite construire un programme qui permettra de retourner à un utilisateur à minima les résultats suivants (vous pouvez rajouter d'autres fonctionnalités si vous le souhaitez) :

- L'utilisateur saisit un nom de produit et une catégorie : le programme doit retourner les produits par notes/ nombre d'avis décroissants ou par prix, selon ce que l'utilisateur aura choisi. S'il y a beaucoup de résultats, n'afficher que les 20 premiers.

Par exemple, si j'écris « Battery » dans la catégorie « Laptop Accessories », alors je pourrais récupérer un résultat tel que « Z55H Battery, Compatible with Sony Headset Battery for WF-1000XM4(2PCS)+Tools »

- L'utilisateur saisit une catégorie, une note minimale et un nombre minimal d'unités vendues: le programme doit retourner les produits correspondants, par prix ou par note.

- **Recommandation de cadeau:** Pour cette partie, le programme devra aider un utilisateur à choisir un cadeau. Pour cela, il devra demander par exemple ses préférences sur la/les catégorie(s) dans lesquelles le programme doit chercher, le budget de la personne, le type de produit qu'il souhaite offrir. Le programme devra ensuite proposer des exemples de cadeaux, en favorisant par exemple les produits Best Seller, bien notés et/ou ayant beaucoup d'avis (on ne voudrait pas recommander des mauvais cadeaux!). Il est également possible d'ajouter d'autres critères, à votre convenance. Le programme doit laisser la possibilité à l'utilisateur de ne pas sélectionner toutes les options.

Par exemple, « je souhaite acheter un cadeau pour bébé pour moins de 100\$, si possible un jouet avec une note moyenne d'au moins 4.5/5 »

Partie 3 : Interface graphique

Une fois le programme de la partie 2 réalisé, l'idée est de ne pas devoir écrire des lignes de code pour exécuter votre programme une fois le script exécuté. Il faudra donc créer une interface afin que l'utilisateur soit guidé pour l'utilisation de votre programme.

Pour ce faire, vous devrez utiliser la librairie Tkinter² pour réaliser une interface graphique. Cette interface devra comporter les éléments suivants:

- Possibilité pour l'utilisateur de saisir les différents critères (selon ce qui sera fait à la partie 2)
- Affichage des résultats selon ce qui a été demandé dans la partie 2 (Idée: incorporer des liens vers les pages des produits sur Amazon et/ou affichage d'images - via les variables `imgUrl` / `productURL`)

Note: en cas de difficultés, il est possible de faire une version plus « simple » de l'interface en demandant les infos à l'utilisateur avec des `input` puis en affichant les résultats dans la console avec `print`. La version avec Tkinter étant plus avancée, elle sera bien sûr plus valorisée au niveau du barème que la version simple de l'interface.

² Quelques références pour Tkinter :

- <https://python.doctor/page-tkinter-interface-graphique-python-tutoriel>
- <https://realpython.com/python-gui-tkinter/>
- <https://www.geeksforgeeks.org/python-gui-tkinter/>