UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE

Master 1 Économétrie, Statistiques
Langage de programmation 1

Rapport Projet Python

Sujet 1 : Amazon Prime Video, Disney + ou Netflix ?

Réalisé par :
BENKIRAN Yasmine
GEFFLOT Claire
PENICHON Romain
GARDONI Adrien

Année universitaire : 2021

Sommaire

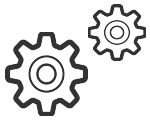
Introduction

À l'issu de l'enseignement Langage de programmation 1 - Python reçu au premier semestre de notre M1 Économétrie, Statistiques, nous sommes amenés à réaliser un projet de groupe. Après une lecture rapide des deux sujets proposés et une courte analyse des données, nous avons décidé unanimement de traiter le sujet 1 : Amazon Prime Video, Disney + ou Netflix ?

Dans le cadre de ce projet, nous avons réaliser un moteur de recherche qui aura pour but d'aider des utilisateurs à choisir quelle serait la plateforme de streaming la plus adaptée en fonction de leurs goûts. Pour cela, nous avions à notre disposition 3 fichiers de données synthétisant le contenu des catalogues de Netflix, Amazon Prime Video et Disney + et également un dernier fichier créé à partir des données « IMDb extensive movies dataset » et qui permet d'avoir, pour les films présents sur l'Internet Movie Database, la note donnée en moyenne par les utilisateurs du site aux films. Nous avons réalisé un premier nettoyage de la data sous Excel pour retirer les incohérences. Par exemple, à travers plusieurs instructions, tous les accents ne faisant pas partie de la langue française ont été automatiquement récrient de la bonne manière. Par la suite, nous avons créé nos DataFrames sur lesquels nous avons travaillé au cours de la réalisation du projet.

La répartition du travail au sein de notre groupe s'est faite de la manière suivante : deux se sont occupés de la partie 1 à savoir l'analyse descriptive des bases et visualisation et les deux autres ont travaillé sur la partie 2 à savoir le moteur de recherche, nous nous sommes enfin retrouvés à 4 pour la création de l'interface (partie 3). Nous avons tout de même au cours de la réalisation de nos premières parties échangé nos avis, nos conseils et nous nous sommes aidés mutuellement.

Notice des instructions pour exécuter le programme



Comment faire pour tourner notre programme ?

Étape 1 : Téléchargez le dossier zip Sujet1_GEFFLOT_BENKIRAN_PENICHON_GARDONI dans C:\Users\TonNom.

Étape 2 : Dézippez le dossier téléchargé à l'étape 1. Ce dernier possède l'arborescence suivante :

Sujet1_GEFFLOT_BENKIRAN_PENICHON_GARDONI
- - DATA
  - - amazon_prime_titles.csv
  - - amazon_prime_titles_Clean.xlsx
  - - Check all data.xlsx
  - - disney_plus_titles.csv
  - - disney_plus_titles_Clean.xlsx
  - - movies_ratings_IMDB.csv
  - - movies_ratings_IMDB_Clean.xlsx
  - - netflix_titles.csv
  - - netflix_titles_Clean.xlsx
- - DATAFRAME
  - - DF DATA.py
- - CODES
  - - INTERFACE
  - - PARTIE_1_QUESTION_1.py
  - - PARTIE_1_QUESTION_2.py
  - - PARTIE_1_QUESTION_3.py
  - - PARTIE_1_QUESTION_4.py
  - - PARTIE_2_QUESTION_1.py
  - - PARTIE_2_QUESTION_2.py
  - - PARTIE_2_QUESTION_3.py
  - - PARTIE_2_QUESTION_4.py
  - - PARTIE1_DESCRIPTION_DATA.py

⚠ Attention :

Vous devez obtenir en dézippant le dossier ce chemin :
C:\Users\TonNom\Sujet1_GEFFLOT_BENKIRAN_PENICHON_GARDONI

Et non celui-ci (erreur fréquente) :

C:
\Users\TonNom\Sujet1_GEFFLOT_BENKIRAN_PENICHON_GARDONI\Sujet1_GEFFLOT_BENKIRAN_PENICHON_GARDONI

Étape 3 : Ouvrez sur Spyder le script DF DATA (ce dernier se trouve dans le dossier DATAFRAME). Exécutez le code en entier.

⚠ Attention :

Après avoir lancer le script DF DATA, et avant de lancer le script INTERFACE, vous devez bien penser à ce que la working directory (cf. en haut à droite de Spyder) soit :
C:\Users\TonNom\Sujet1_GEFFLOT_BENKIRAN_PENICHON_GARDONI\CODES

Étape 4 : Ouvrez sur Spyder le script INTERFACE (ce dernier se trouve dans le dossier CODES). En exécutant le code, vous allez accéder à l'interface.

Étape 5 : Ouvrez sur Spyder les différents scripts les uns après les autres en suivant la chronologie (PARTIE_1_QUESTION_1 → PARTIE_1_QUESTION_2 → … → PARTIE_2_QUESTION_4). Exécutez à chaque fois le code et découvrez les réponses aux questions. Vous pouvez découvrir les graphiques de la partie 1 et vous prêtez au jeu en entrant des inputs dans la partie 2.

Présentation du jeu de données

Dans un premier temps, nous avons effectuer une description classique du jeu de données. Notre curiosité a tout d'abord porté sur la taille des fichiers (nettoyés) pour nous donner une idée sur la quantité de données que nous allons manipuler. La taille du fichier Amazon Prime (vous trouverez ce dernier à l'adresse suivante : 'C:\\Users\\TonNom\\Sujet1_GEFFLOT_BENKIRAN_PENICHON_GARDONI\\DATA\\amazon_prime_titles.csv') est de 3 972 416 octets. La taille du fichier Netflix (vous trouverez ce dernier à l'adresse suivante : 'C:\\Users\\TonNom\\Sujet1_GEFFLOT_BENKIRAN_PENICHON_GARDONI\\DATA\\netflix_titles.csv') est de 3 399 671 octets. Et la taille du fichier Disney + (vous trouverez ce dernier à l'adresse suivante : 'C:\\Users\\TonNom\\Sujet1_GEFFLOT_BENKIRAN_PENICHON_GARDONI\\DATA\\disney_plus_titles.csv') est de 362 829 octets. Le fichier le plus lourd est le fichier Amazon Prime, suivi de près par celui de Netflix et enfin le dernier est celui de Disney +.

Dans un deuxième temps, les résultats récupérés dans la réalisation des différentes questions de la partie 1 donnent des indications pertinentes sur le jeu de données. En effet, la réalisation de la question 3 apporte cette informations telles que la quantité de films et séries proposés par les différentes plateformes de streaming, ce qui est à prendre en compte pour le choix de l'utilisateur. Amazon Prime Video propose 9 668 films et séries, Netflix 8 808 et Disney + 1 368. Amazon Prime Video présente donc le plus de films et séries. En revanche, Amazon Prime Video propose 7 814 films et 1 854 séries, tandis que Netflix propose 6 132 films et 2 676 séries. Cela peut être également pris en compte dans le choix de la plateforme la plus adéquate pour l'utilisateur qui peut privilégier un type de contenu (films ou séries). Si on associe ce critère aux genres les plus proposés des plateformes, on peut avoir une base très solide pour analyser des choix plus précis.

La question 4 de la partie 1 nous demande la note moyenne des films par genre pour chaque plateforme. Pour choisir une plateforme de streaming adéquate, l'utilisateur qui a en tête un genre en particulier, peut décider d'aller les visionner sur la plateforme qui possède les notes moyennes pour le genre en question les plus élevées car cela est un gage de qualité. Par exemple, la note moyenne des films classés dans drama sur Netflix est de 6.17 tandis que pour Amazon Prime Video, cette moyenne est de 5.93. On constate que Netflix obtient pour ses genres de films les plus représentés de meilleures notes moyennes pour 4 catégories sur 5 par rapport à Amazon Prime. Disney + obtient des notes moyennes supérieures à 6.00 pour ses 5 genres les plus représentés, avec pour son genre documentary, la note moyenne la plus élevée des genres les plus représentés des 3 plateformes avec une moyenne de 6.725. Ces résultats associés à ceux de la question précédente permettent à l'utilisateur d'appliquer des filtres efficaces en fonction de ses préférences pour choisir une plateforme de streaming adéquate.

Partie 1 : Analyse descriptive des bases et visualisation

Question 1 :

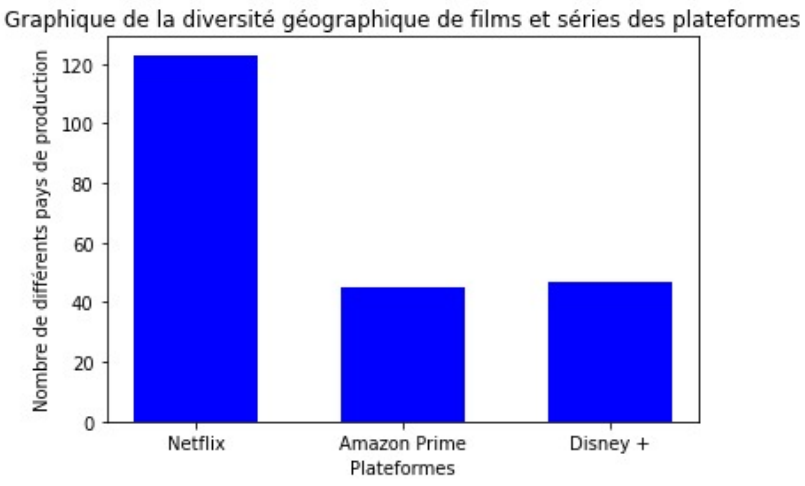
• Quelle est la plateforme qui contient le plus de diversité géographique de films et séries ? Faire un graphique ?

Afin de répondre à cette question, nous avons adopté le raisonnement suivant : pour obtenir la diversité géographique des films et séries des différentes plateformes et pouvoir mener une comparaison, il fallait que nous puissions trouver pour chacune des plateformes le nombre de pays de production différents dans lesquels ont été tournés les films et séries.

Dans cet objectif, nous avons dans un premier temps pour chacune des plateformes créé des Séries contenant une colonne des pays de production (« country ») à partir du DataFrame* de la plateforme concernée qui a été créée au tout début de la réalisation de notre projet (contenant les données « nettoyées »). Par exemple, pour la plateforme de streaming Netflix, les Séries créées sont df_netflix_country, et pour les créer nous avons utilisé la ligne de code suivante df_netflix_country = df_netflix.country. Ensuite, toujours pour chacune des plateformes, nous avons créé des listes que nous avons « nettoyé » et arrangé de sorte à obtenir une liste finale avec tous les pays de production différents (sans incohérences, sans doublons, etc.). Et nous avons calculé la taille des différentes listes. La taille de la liste de la plateforme Netflix est 123, d'Amazon Prime Video 45 et de Disney + 47. Autrement-dit, le nombre de pays de production différents pour la plateforme Netflix est de 123, pour Amazon Prime Video de 45 et enfin pour Disney + de 47. Enfin, nous avons créé un dictionnaire dans le but de réaliser le graphique en barres (un histogramme). Ce type de graphique facilite la comparaison.

La plateforme de streaming qui contient le plus de diversité géographique de films et séries est donc Netflix.

*Un DataFrame est un tableau bi-dimensionel, mutable et pouvant contenir des données de types hétérogène et tabulaire. Ses lignes et colonnes sont labelisées.



Question 2 :

• Quelle est l'évolution au cours du temps de l'offre (i.e. nombre) de films et séries pour chacune des plateformes de streaming ? Faire la distinction Amazon Prime/Netflix/Disney + et également une distinction film/série.

Afin de répondre à cette question, nous avons adopté la démarche suivante :

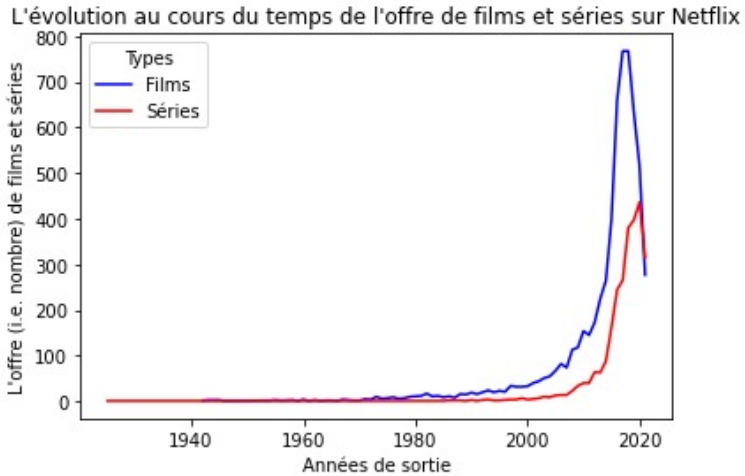
- Étape 1 : Nous avons créer pour la plateforme de streaming Netflix un DataFrame comprenant seulement les films de Netflix. Puis nous avons compté le nombre de films sortis chaque année. Et créer des Séries avec les années de sortie des films et le nombre de films sortis ces années en question.

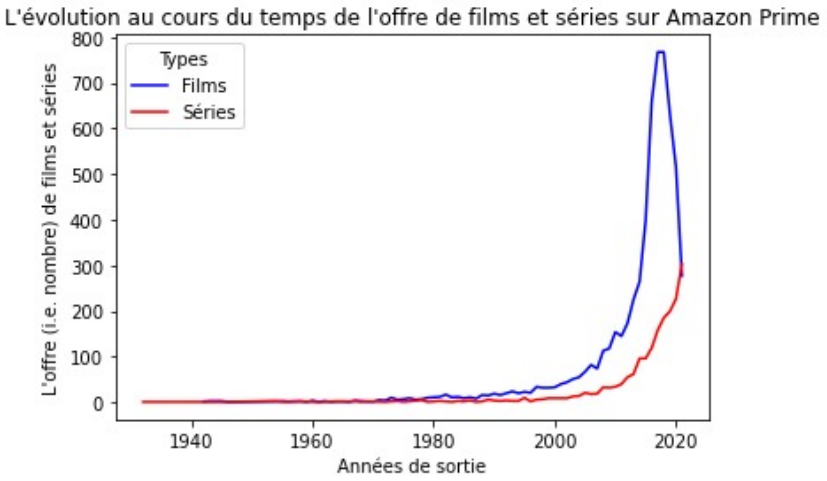
- Étape 2 : Nous avons reproduit l'étape 1 pour les séries de la plateforme Netflix.

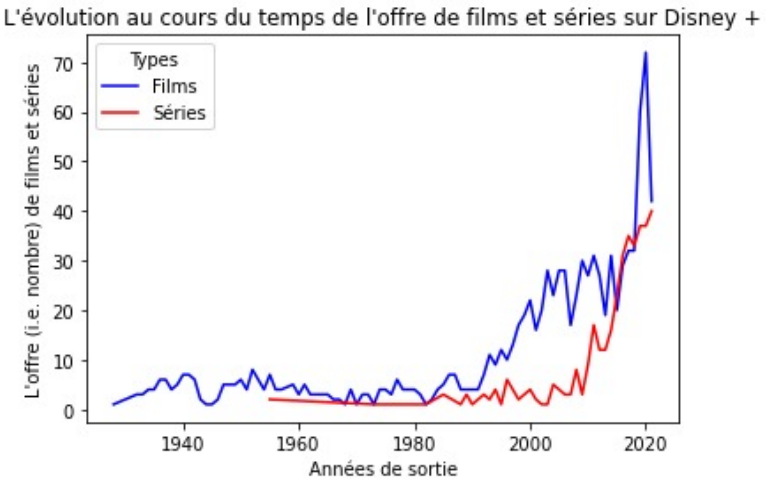
- Étape 3 : Création du graphique de l'évolution au cours du temps de l'offre (i.e. nombre) de films et séries sur Netflix. Le graphique comporte deux courbes, une représentant les films et l'autre les séries.

- Étape 4 : Nous avons reproduit les trois mêmes étapes pour les plateformes Amazon Prime Video et Disney +.

Nous pouvons conclure cette question en interprétant les graphiques obtenus à l'issu des étapes précédentes : l'évolution au cours du temps de l'offre (i.e. nombre) de films et séries connaît une tendance à la hausse pour les trois plateformes de streaming. Cependant, les évolutions sont plus ou moins différentes selon les plateformes. En effet, pour Netflix, l'offre de films et séries est nulle jusqu'aux années 80 avant de connaître une augmentation exponentielle jusqu'en 2020, et une chute fulgurante jusqu'à présent. Pour Amazon Prime Video, les premiers films et séries ont été créés de même qu'à partir des années 80, le nombre de films et séries a augmenté de manière exponentielle. Cependant, en 2020 le nombre de films sortis a fortement diminué (divisé par environ 2,5), quant aux séries l'évolution est toujours positive. Enfin pour Disney +, l'évolution au cours du temps de l'offre de films et séries a beaucoup fluctué. Le nombre de films réalisés est supérieur au nombre de séries pour chacune des plateformes de streaming.







Question 3 :

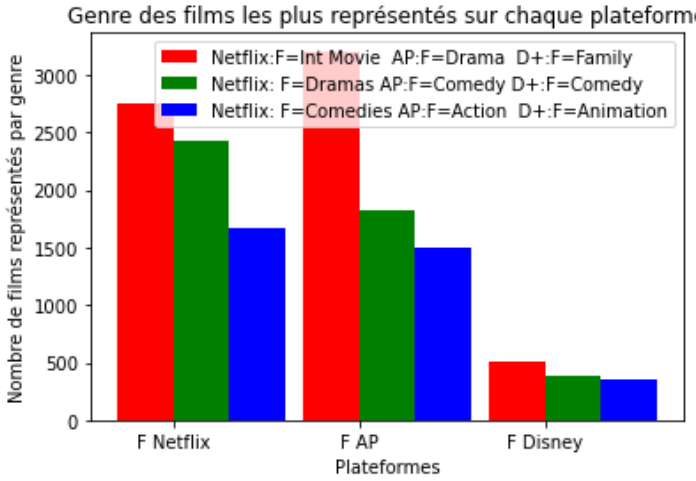
• Quels sont les genres de films/séries les plus représentés sur chaque plateforme de streaming ? Faire un graphique global/films uniquement/séries uniquement.

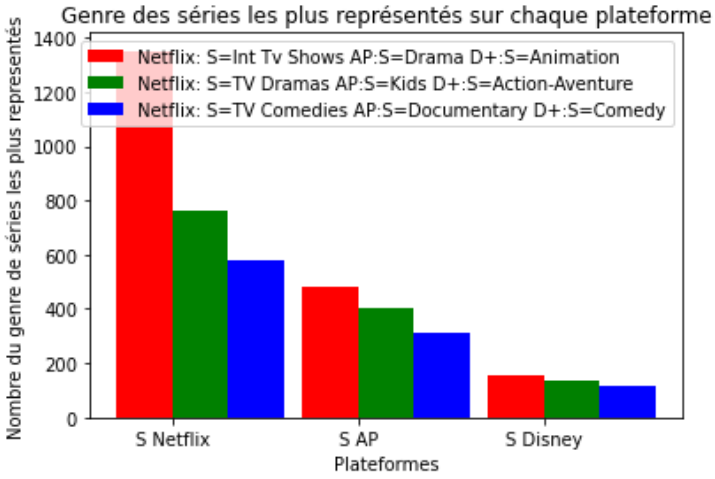
Dans le cadre de l'analyse des résultats de la question 3 de la partie 1, il est pertinent de reprendre les éléments énoncés dans la présentation du jeu de données. En effet, la réponse à cette question nous indique les genres les plus représentés pour chacune des plateformes de streaming. Les résultats nous montrent que Pour Amazon Prime et Netflix, les genres les plus représentés pour les films et les séries ne sont pas très différents, des genres comme comédies ou drama se classent a peu près aux mêmes occurrence pour les 2 plateformes pour les films. Pour les séries, on retrouve encore ces 2 genres, avec une présence plus forte des programmes pour enfants. On retrouve pour Disney + des différences avec les autres plateformes concernant les genres les plus représentés, car Disney + cible un public différent avec pour ses genres les plus représentés des catégories comme « Family » ou « Animation », que l'on retrouve dans une proportion bien plus faible sur les autres plateformes. Ces informations, couplées à la présentation du jeu de données, peuvent déjà permettre à l'utilisateur de s'appuyer sur une base solide dans son choix de plateforme la plus adéquate. Sa préférence entre la quantité de films ou de séries

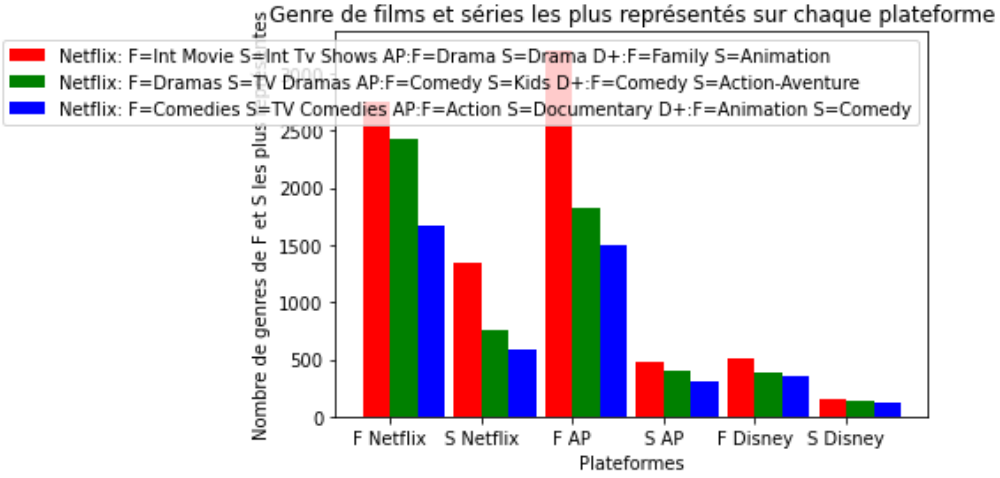
d'une plateforme, et le genre qu'il recherche en majorité pour chacune d'entre elle, sont des aspects que les résultats de cette question permettent d'appuyer.

Graphiquement, il est très pertinent pour l'utilisateur de s'intéresser aux représentations visuelles des types de contenus les plus représentés par genre. Que ce soit le contenu global, ou seulement les films ou les séries, on peut rapidement apercevoir les différences entre les plateformes. Tout d'abord, les quantités de contenus par genre permettent une comparaison rapide, il est par exemple évident de voir la différence de quantité de contenu entre Disney + et les autres plateformes. On représente ici pour plus de clarté, les 3 genres les plus représentés pour chaque plateforme, on a vu précédemment qu'Amazon Prime proposait plus de films que Netflix et que Netflix proposait plus de séries qu'Amazon Prime. C'est également dans ces domaines que les distinctions entre les deux plateformes sont les plus représentatives. Les représentations graphique des genres les plus représentés de ces deux plateformes ont des proportions plutôt similaires, sauf pour le genre le plus représenté de film pour Amazon Prime et le genre le plus représenté de séries pour Netflix qui ont une bien plus grande part que pour les seconds et troisièmes genres les plus représentés.

Nous avons rencontrés certains problèmes que nous allons vous exposer. En, effet, en suivant le raisonnement énoncé plus tôt, nous avons tout d'abord commencé par utiliser des « .loc » sur les DataFrames des plateformes pour récupérer les colonnes qui nous intéressaient. En appliquant des conditions à ces DataFrames, et en utilisant l'instruction Counter, nous sommes parvenus à un résultat possiblement juste, mais très imprécis et incomplet. Nous n'avions pas réussi à séparer les genres, cela a été la plus grande difficulté. La méthode Counter était également imprécise, car nous devions regarder manuellement les genres les plus représentés, laissant possiblement place à des erreurs. Afin de corriger cela, nous avons dû individualiser les genres des films et séries comprenant plusieurs genres, en mettant le tout dans des listes. Nous avons ensuite pu compter les occurrences des genres, mais nous ne parvenions pas à séparer les films des séries, donc encore une fois, nous n'arrivions pas à réunir toutes les informations que nous avions récolté pour pouvoir en tirer une conclusion. Pour surmonter cela, nous avons dû rajouter une étape à nos conditions souhaitées, à savoir films ou série, et de rajouter les plateformes. Nous n'avions pas encore essayé de poser plusieurs conditions, en reprenant donc ces infos et en comptant incorporer les genres séparés, nous avons pu compter les occurrences pour les séries et films, par plateforme, par genre et en prenant en compte les 5 occurrences de genre les plus élevées pour chaque genre.







Question 4 :

• Quelle est la note moyenne des films par genre pour chaque plateforme ? Se concentrer uniquement sur les 5 genres les plus représentés pour créer une représentation graphique des notes moyennes par genre, par plateforme.

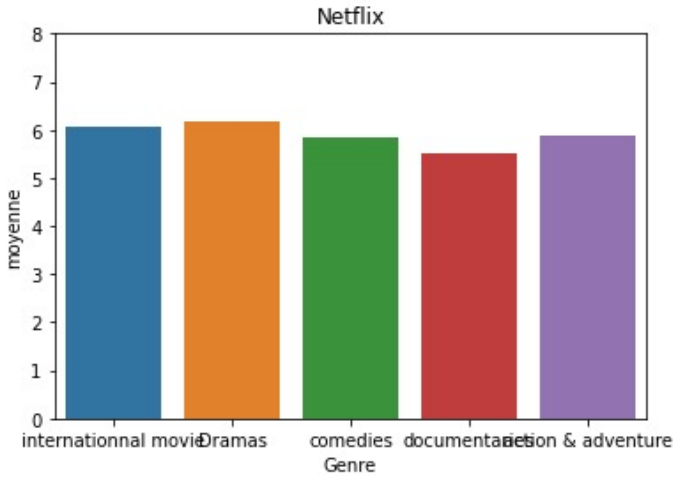
Pour répondre à cette question, le code prendra la forme suivante : nous allons dans un premier temps prendre les lignes avec le genre qui nous intéresse dans les DataFrames de nos

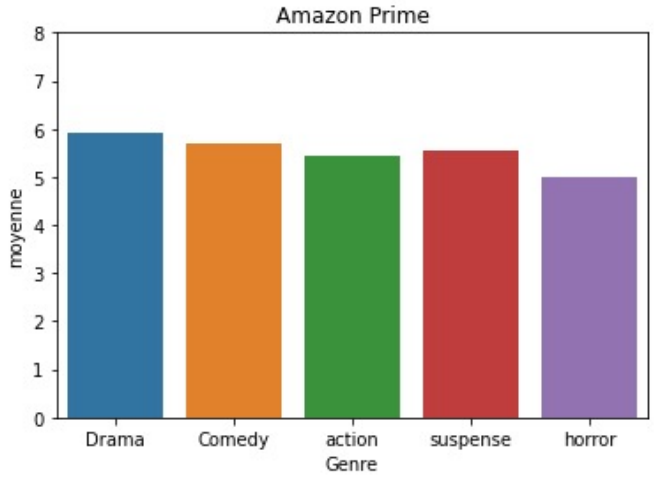
plateformes. Ensuite, on sélectionne les colonnes « title » et « genre » pour chaque plateforme et « title » et « avg_vote » pour imdb. Nous allons par la suite indexer « title » pour chaque plateforme avec imdb et joindre chaque plateforme avec imdb. L'étape suivante consiste à faire des listes avec des avg_vote, puis de supprimer les « nan » de ces listes. Pour finir, nous réalisons la moyenne des éléments des listes.

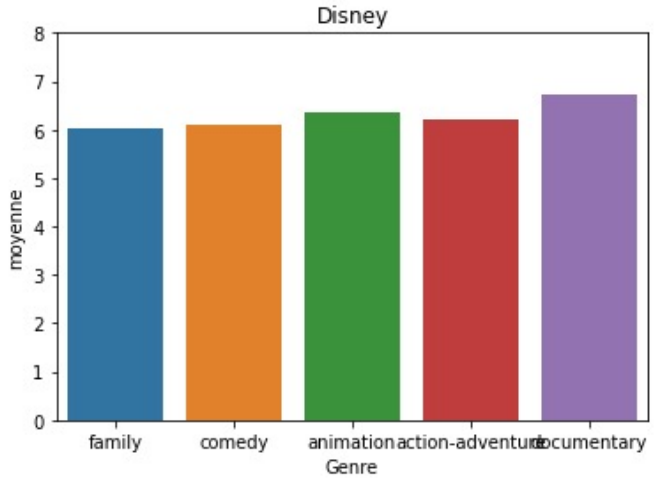
Cette question nous demande la note moyenne des films par genre pour chaque plateforme. Pour choisir une plateforme de streaming adéquate, un utilisateur qui choisit un genre en particulier, peut décider d'aller les visionner sur la plateforme où les notes moyennes pour le genre en question sont les plus hautes. Par exemple, la note moyenne des films classés drama sur Netflix est de 6.17 tandis que pour Amazon Prime Video, cette moyenne pour le même genre est de 5.93. On constate que Netflix obtient pour ses genres de films les plus représentés de meilleures notes moyennes pour 4 catégories sur 5 par rapport à Amazon Prime Video. Disney + obtient des notes moyennes supérieures à 6.00 pour ses 5 genres les plus représentés, avec pour son genre le plus représenté qui est documentary, la note moyenne la plus élevée des genres les plus représentés des 3 plateformes avec une moyenne de 6.725. Ces résultats associés à ceux de la question précédente permettent à l'utilisateur d'appliquer des filtres efficaces en fonction de ses préférences pour choisir une plateforme de streaming adéquate .

Ces résultats sont illustrés graphiquement, mais ne permettent pas de dégager une tendance évidente sans l'analyse effectuée plus tôt. Les écarts entre les notes n'étant globalement pas très élevés, toutes les colonnes des graphiques sont assez similaires.

Les difficultés auxquelles nous avons été confrontés pour cette question ont été multiples. Contrairement à la question précédente, un nouveau DataFrame comprenant les notes de films doit être pris en compte. Ce dernier n'est pas au même format que les DataFrames des plateformes, et possède des colonnes différentes. Cette question nous a confronté à plus de problèmes que la précédente, car notre idée initiale, concernant le schéma à suivre pour extraire les informations souhaitées, nous a amené à beaucoup d'erreurs en retour console. La fusion des DataFrames globaux des plateformes avec celui des notes ne nous permettait pas d'extraire des colonnes avec des conditions. Pour tenter de résoudre ces problèmes, nous avons donc opté pour la division de notre réponse en prenant les 5 genres les plus représentés pour chaque plateforme, et en appliquant des conditions qui fonctionnaient pour un genre ciblé.


Netflix
moyenne
Genre
internationnal movie
Dramas
comedies
documentaries
action & adventure


Amazon Prime
moyenne
Genre
Drama
Comedy
action
suspense
horror


Disney
moyenne
Genre
family
comedy
animation
action-adventure
documentary

Partie 2 : Moteur de recherche

Question 1 :

• L'utilisateur saisit le titre d'un film ou d'une série : Afficher les plateformes sur lesquelles le film/la série est disponible avec des informations comme l'année de sortie, le synopsis, les acteurs principaux, le genre. Si c'est un film, donner sa note sur IMDb si elle existe. Prendre en compte le cas où l'utilisateur entre un film qui n'existe dans aucun catalogue.

Afin de répondre à cette question, nous avons utilisé une fonction qui donne les caractéristiques du titre recherché.

L'input est le suivant :

-- title: str, l'utilisateur entre son choix de film ou série

L'output est le suivant :

— subset_df_global_title: dataframe, nom du film ou série avec ses caractéristiques

Nous avons cependant principalement rencontré pour cette question des problèmes au niveau des jointures. Ainsi que pour la clarté et la synthétisation des informations au sein de la fonction. L'idée principale était de rassembler au sein d'un même DataFrame le dossier des notes de films ainsi que celui des plateformes et d'appliquer une fonction avec pour seul DataFrame, celui que nous aurions fusionné et sur lequel nous pourrions poser des conditions. Cette approche n'a pas fonctionné, nous avons obtenu des erreurs au sein de la console ainsi que des données inexploitables pour la plupart. Nous avons donc envisagé une jointure entre les deux DataFrames, séparés cette fois, afin de pouvoir demander le titre dans ma fonction, et d'obtenir les informations qui découlent de nos DataFrames. Nous avons également du optimiser la taille de notre code pour rendre la fonction la plus claire possible, ce qui s'est amélioré après plusieurs essais.

Question 2 :

• L'utilisateur entre le nom d'un acteur : Suggérer une plateforme entre les 3 en fonction du nombre de films/séries de l'acteur disponible sur chacune des plateformes, puis afficher les films/séries de cet acteur disponibles avec des informations dessus (type, nom, année, synopsis, classification et genres).

Afin de répondre à cette question, nous avons utilisé une fonction qui donne les titres de films ou séries (avec ses caractéristiques) de l'auteur recherché, et le nombre de fois qu'il apparaît dans chaque plateforme.

L'input est le suivant :

-- acteur: str, l'utilisateur entre le nom de l'acteur préféré

L'output est le suivant :

-- subset_df_global_actor_colonne : dataframe, nom du film ou série avec ses caractéristiques dans lequel l'acteur apparaît

-- count_actor.title: Int64, nombre de films ou séries dans lequel l'acteur apparaît par plateforme

Nous avons rencontré cependant certains difficultés dans la réalisation de cette question. En effet, nous avons collectivement eu plusieurs échecs avant de réussir à trier nos données en les séparant. Pour les genres, les pays, et ici les acteurs, parvenir à récolter toutes les occurrences de ces derniers sachant qu'ils sont répartis par groupe d'acteurs nécessitait de trouver un séparateur au sein des DataFrames. Ici, les virgules séparent le nom des acteurs, nous avons donc pu surmonter ce problème en appliquant cette condition dans notre fonction. Il fallait ensuite assembler les éléments demandés dans la fonction de manière ordonnée, en comprenant vraiment ce qu'on cherchait à savoir pour pouvoir appliquer les bonnes conditions aux DataFrames. Cela nous a demandé de nombreuses tentatives avant de parvenir au résultat souhaité.

Question 3 :

• L'utilisateur saisit un type de contenu (film/série) et un genre : Afficher le nombre de résultats qui remplissent les critères d'entrée pour chacune des plateformes puis parmi ces résultats les 3 films ou séries les plus récent(e)s pour chaque plateforme.

Afin de répondre à cette question, nous avons utilisé une fonction qui donne le nombre de résultats remplissant les critères d'entrée pour chacune des plateformes puis parmi ces résultats les 3 films/séries ajoutées le plus récemment pour chaque plateforme.

Les inputs sont les suivants :

-- typ: str, l'utilisateur donne sa préférence entre movie ou tv show

-- genre: str, l'utilisateur donne sa préférence pour un genre de film/série

Les outputs sont les suivants :

-- count_genre_type.title: dataframe, nombre de résultats qui remplissent les critères d'entrée par plateforme

-- most_recent_netflix: dataframe, les 3 titres de films/séries ajouté(e)s dernièrement par Netflix

-- most_recent_AP: dataframe, les 3 titres de films/séries ajouté(e)s dernièrement par Amazon Prime

-- most_recent_disney: dataframe, les 3 titres de films/séries ajouté(e)s dernièrement par Disney +

Lors de cette question nous avons fait face à diverses problématiques comme la création d'un subset de notre DataFrame prenant en compte deux conditions (le type et le genre) ainsi que les

virgules présentent au sein de la catégorie genre (listed_in) de la DataFrame. La fonction str.contains nous a notamment permis de passer outre cette problématique sans avoir à séparer dans une autre DataFrame les auteurs (exemple : en utilisant la fonction split). Ensuite nous avons créé trois autres subset DataFrame par plateforme selon les deux critères d'entrée. Enfin, nous avons trié ces DataFrames par dates d'ajouts ce qui nous a permis d'avoir en output le nombre de films disponibles correspondant à nos conditions (type et genre) par plateforme, ainsi que les trois titres de films/séries ajouté(e)s dernièrement sur chaque plateforme.

Question 4 :

• Demander à l'utilisateur des informations sur ses préférences : séries ou films, genre, rating, pays de production et éventuellement le nom d'un ou de plusieurs acteurs et suggérer du contenu sur chaque plateforme. Pour cette suggestion, vous pouvez par exemple donner les 5 films/séries les plus récent(e)s pour chaque plateforme, les 5 films/séries ajoutées le plus récemment etc. Vous mentionnerez le critère choisi.

Afin de répondre à cette question, nous avons utilisé une fonction qui donne les 5 films/séries ajoutées le plus récemment ainsi que les 5 films/séries les plus récents pour chaque plateforme selon les préférences de l'utilisateur.

Les inputs sont les suivants :

-- typ_pref: str, l'utilisateur donne sa préférence entre movie ou tv show
-- genre_pref: str, l'utilisateur donne sa préférence pour un genre de film/série
-- rating_pref: str, l'utilisateur donne sa préférence pour la classification d'un(e) film/série
-- actor_pref: str, l'utilisateur donne un nom d'un ou plusieurs acteurs
-- country_pref: str, l'utilisateur donne sa préférence pour un pays de production

Les outputs sont les suivants :
En combinant de 1 à 5 préférences, selon les critères d'entrée

-- condition_ry: dataframe, les 5 titres de films/séries les plus récent(e)s par plateforme
-- condition_dat: datframe, les 5 titres de films/séries ajouté(e)s dernièrement par plateforme

Cette question a été la plus complexe à réaliser car elle combinait l'ensemble des questions de la partie 2 et nous avons notamment eu des problèmes de syntaxes pour l'utilisation de la fonction ELIF d'où notre fonction avec des if en escalier.

Tout d'abord, nous avons posé le problème à l'écrit pour nous aider à mieux visualiser ce que nous voulions réellement en output et comment y parvenir. Nous sommes partis du premier cas où l'utilisateur nous donne les 5 conditions, et nous avons donc créé notre subset dataframe all_conditions. Ensuite, nous avons trié notre DataFrame all_conditions par ordre de sortie (release_year) et par ordre d'ajout (date_added) et nous avons donc ainsi créé deux autres DataFrames.

De plus, notre autre problématique était de sortir uniquement les cinq titres de films/séries les plus récent(e)s par plateforme et les 5 titres de films/séries ajouté(e)s dernièrement par plateforme. Pour cela nous avons « ranké » nos deux DataFrames (all_conditions_sorted_release_year et all_conditions_sorted_date_added) par plateforme du plus récent au plus ancien.

Enfin, nous avons ajouté la colonne rank à nos deux DataFrames et nous avons gardé uniquement les 5 premiers films/séries par plateforme. Nous avons effectué cette démarche pour différents cas lorsque l'utilisateur nous fournit uniquement quatre préférences, trois préférences, deux et une seule préférence. Par exemple, si l'utilisateur nous donnait que trois préférences nous conservions uniquement le type, le genre, et la classification et nous retirions de notre fonction acteur et pays de production (choix arbitraire).

Par manque de temps, nous n’avons pas pu creuser la partie où notre fonction prend en compte l'ensemble des combinaisons possibles concernant ce que l'utilisateur nous donne en input. Mais nous pensons que cela aurait été intéressant de créer une classe et une sorte de boucle qui nous prédit l'ensemble des combinaisons possibles. Pour finir, cette question a été pour nous la plus intéressante et la plus enrichissante en terme de compétences techniques.

Partie 3 : Interface

Pour la modélisation de l'interface, nous avions fait le choix d'utiliser Tkinter car nous avions pour objectif de proposer à l'utilisateur un moteur de recherche un peu plus poussé que la version simple dans lequel il serait possible d'afficher les résultats recherchés sur une interface simple, et sans l'apparition de code effectué en amont.

Tout d'abord, nous avons importé de nouveaux modules que nous n'avions jamais utilisé auparavant tel que : Tkinter (qui permet de personnaliser un interface), Webbrowser (qui permet d'ouvrir des liens internet à partir des codes). De plus, nous avons importé les modules de la partie 2 du projet afin qu'on puisse utiliser les fonctions dans la programmation du moteur de recherche sur l'interface.

Nous avons par la suite créé l'interface principale dans lequel nous avons personnalisé et ajouté des widgets qui permettront par la suite de faire appel aux fonctions créées en amont du script (command=fonction), bien évidemment nous avions pu personnaliser ces widgets.

Ensuite, nous avions créé de nouvelles fonctions permettant d'effectuer les commandes souhaitées, tels que cliquer sur un bouton qui nous dirige sur la page internet voulu, se diriger sur un nouvel onglet afin d'écrire une recherche voulu sur le moteur de recherche et dans lequel le résultat obtenu s'affiche sur ce même onglet. Pour certaines fonctions nous avons été contraints de créer des sous-fonctions dans le but de lancer les fonctions des modules importés. Grâce à ces sous-fonctions nous avons pu afficher les résultats rechercher par l'utilisateur de la même manière que si l'utilisateur aurait lancer sa recherche en donnant l'input dans la version simple.

Nous avons fait le choix de mettre les fonctions en amont car Python ne peut pas lancer les codes demandés sur l'interface principale si les fonctions n'ont pas été déterminées en amont. Cela n'empêche pas de créer en premier lieu l'interface principale puis de créer les fonctions en second lieu.

Ensuite nous avons créé un menu dans lequel chaque option a pour but d'ouvrir un onglet en fonction du type de recherche voulu. Chaque onglet est composé d'un ou plusieurs moteur(s) de recherche dans lequel l'utilisateur inscrira les informations recherchées. Nous avons aussi ajouté un bouton « Quitter » qui a pour but de fermer l'interface principale et toutes les pages ouvertes à partir de cette interface.

À travers la modélisation de l'interface, nous avons principalement rencontré des problèmes au niveau de la mise en marche des fonctions. Il fallait commenter les inputs que nous avons fait dans la partie 2 et mettre en argument dans nos fonctions directement (fonctions qui sont les modules importés). Nous avons ensuite supprimer l'input au sein des scripts Python afin de les intégrer en argument dans nos fonctions afin de prélever la saisie dans la barre de recherche de l'interface.

Puis, le diriger directement vers la sous-fonction importée. Nous avons du mettre les DataFrames en argument également, afin de nous éviter de créer une fonction particulière. Nous avons également rencontré des problèmes sur la forme au sein de notre interface, pour la présentation des résultats. La clarté des informations sorties n'est pas optimale. Nous n'avons pas pu surmonter ce problème par manque de temps. On supposerait donc que le résultat affiché est directement issue du résultat affiché dans la console. Nous aurions donc aimé donner un résultat plus précis sous forme de tableau.

Prolongements et applications possibles

Les résultats précédemment trouvés concernant le contenu des plateformes, sont interprétables de manière plus pertinente en prenant en compte des aspects qui ne sont pas clairement donnés par la data des plateformes. Pourquoi insister sur les comparaisons entre Amazon Prime Video et Netflix ? Comment Disney + pourrait être une plateforme adéquate pour un utilisateur malgré un contenu plus réduit ? L'une des principales réponses est que Disney + cible un public bien plus précis que les autres plateformes, et qu'elle ne propose que ses propres licences. Pour Disney + , les genres sont orientés vers une tranche d'âge plus faible et sont donc moins variés que pour les autres plateformes, car le public visé ne correspond qu'à des genres en accord avec l'image de Disney. Il peut donc être pertinent d'opter pour cette plateforme si le principal motif recherché est le genre. Proposé en France depuis 2014, Netflix a eu un quasi-monopole des plateformes de streaming jusqu'en 2020. Les parts de marché étaient très importantes, et son expansion ne cessait de croître. Avec l'apparition d'Amazon Prime Video, Netflix voit apparaître un concurrent direct qui propose d'autres avantages que le streaming et qui a les moyens de grandir de manière importante. L'apparition de Disney + a également mis un coup de frein aux parts de Netflix, car le coût des plateformes de streaming nécessite de privilégier un nombre restreint d'abonnements.

Pour tenter d'aller plus loin dans le choix d'une plateforme, nous pourrions prendre en compte des possibilités de choix qui ne sont pas proposées dans les questions. Par exemple, pour Amazon Prime, est ce que l'utilisateur commande régulièrement sur Amazon et qu'un abonnement de livraison Prime fourni avec lui sera plus profitable ? Ou bien est ce que l'utilisateur aime le football français ? Car il est possible de regarder les matchs de Ligue 1 sur Amazon Prime Video en rajoutant un supplément dans l'abonnement Prime Video. Pour Netflix, nous pourrions prendre en compte le nombre conséquent de séries et films à succès créés par la plateforme, et qui peut pousser un individu à l'abonnement pour une quantité de programmes réduite, mais qu'il souhaite regarder en particulier. Pour Disney +, il faudrait prendre en compte le fait qu'un individu qui répond au moteur de recherche peut potentiellement le faire pour chercher un programme qui plaise à son ou ses enfant(s). Dans ce cas, ce n'est pas nécessairement l'utilisateur qui pait l'abonnement qui va regarder les programmes.

Enfin, en application directe à notre sujet, on aurait pu dans la partie 2 question 4 pousser davantage le code pour avoir l'entièreté des combinaisons possibles si l'utilisateur précise deux ou trois ou quatre préférences. Nous aurions également pu davantage pousser l'interface Tkinter en intégrant par exemple dans notre code des classes, en rajoutant des boutons, en améliorant design, etc. Nous aurions aimé réaliser toutes ces modifications mais le manque de temps nous a contraint à y renoncer.

Conclusion

Qu’avons-nous appris ?

Yasmine : La réalisation de ce projet a sans doute été pour moi la meilleure façon d’apprendre efficacement à programmer en Python. Ma progression a été exponentielle. En effet, mon implication et mes longues recherches acharnées de solutions à mes problèmes, m’ont permis de m’améliorer considérablement. J’ai également eu l’occasion de valider mes connaissances assimilées au cours du module de programmation Python, mais aussi de les approfondir. Contrairement aux travaux dirigés effectués en classe, le projet m’a contraint à apprendre à structurer ma pensée, créer un cheminement de réponse. Mais également, à être extrêmement patiente et attentive dans la recherche des réponses. Enfin, j’ai apprécié travailler avec les membres de mon groupe, nos échanges on été très intéressants et enrichissants.

Claire : J’ai beaucoup apprécié réaliser ce projet Python. Pratiquer m’a permis de progresser de manière fulgurante. J’ai pu utiliser tout au long de la réalisation du projet les connaissances acquises durant le module de programmation Python, en les approfondissant. En effet, mes longues heures de recherche et les difficultés que j’ai pu rencontrer m’ont permis de développer mes compétences. J’ai appris entre autres à structurer mon raisonnement, à raccourcir mes codes en utilisant des fonctions de manière efficace et optimale. Je tiens enfin à remercier les membres de mon groupe pour nos échanges très instructifs.

Romain : Ce projet a été pour ma part le moyen le plus efficace de progresser en programmation ce semestre. Il regroupe plusieurs aspects, plusieurs dimensions et cela m’a permis d’apprendre beaucoup d’éléments. Tout d'abord le code en lui-même, le fait que ce projet repose sur des questions donne une direction à suivre, contrairement à une problématique à laquelle répondre, la contrainte des questions fait que l'on va potentiellement plus bloquer sur des problèmes. Et sur un sujet plus libre où l'on pourrait trouver un autre moyen, ici on progresse en cherchant comment résoudre ce problème. C'est comme cela que j'ai pu apprendre à réaliser des fonctions ordonnées sur des valeurs que j'ai pré-sélectionné de manière à répondre efficacement. Je trouve avoir vraiment progressé dans la manière de réduire mes codes pour aller à l'essentiel grâce aux recherches effectuées et aux possibilités de codes découvertes durant ce projet. Le côté analyse, interprétation, et conclusion, qui sont tout aussi important que la partie code, m'ont permis pour la première fois de travailler de manière complète sur un projet de programmation.

Adrien : Ce projet a été une très bonne mise en pratique des connaissances acquises au cours du semestre, il m'a permis de répondre à des questions d'analyse et de traitement de données, de synthétiser des résultats, et de tirer des conclusions pertinentes sur un sujet. En terme de code pur, j'ai réalisé que je pensais maîtriser des thèmes qui en réalité m'étaient encore flous. Notamment sur la création de graphique, qui nécessite une réelle compréhension des données exploitées, et de cerner ce qu'on veut faire dire à notre résultat pour l'illustrer au mieux

visuellement. J'ai également appris a trouver des solutions par moi même, en cherchant sur internet. En bloquant sur beaucoup de points j'ai du trouver meilleure manière de s'en sortir. En cherchant la solution d'un problème sur internet, on apprend énormément de choses qui ne servent peut être pas sur le moment, mais dont on découvre l'existence ou l'utilité ce qui permet par la suite d'engendrer un bagage de connaissances que l'on aurait potentiellement pas acquérir en trouvant le solution plus rapidement. J'ai également appris a mieux maitriser panda, travailler avec des DataFrames permet de comprendre et différencier chaque partie de stockage, en fonction de ce que l'on souhaite faire. Une liste et un DataFrame peuvent être similaire mais on n'effectuera pas les mêmes actions dessus. Je comprends donc beaucoup mieux les facteurs qui amèneront à décider par quelle méthode de classification des données on va procéder. Notamment en tentant de créer des listes pour la question 3 partie 1 qui n'était pas la bonne méthode dans mon approche.