



# **Langage de programmation 1**

Projet Python

Analyse de données Amazon US

# Sommaire

<b>Introduction .....</b>	<b>2</b>
<b>I / Analyse descriptive et visualisation .....</b>	<b>2</b>
<b>II / Recherche de contenu .....</b>	<b>3</b>
<b>III / Interface graphique .....</b>	<b>4</b>
<b>IV / Difficultés rencontrées .....</b>	<b>6</b>
<b>V / Apprentissage via le projet.....</b>	<b>8</b>

# Introduction

Dans un premier temps, ce projet a consisté en la réalisation de programmes visant à analyser les données d'une base de données de produits Amazon US. Pour ce faire, nous avons principalement utilisé la bibliothèque pandas. Puis, la seconde étape du projet fut la réalisation d'une interface graphique par le biais de la bibliothèque Tkinter.

Ce rapport permettra de rendre compte en détail des démarches que nous avons entreprises afin de réaliser nos programmes. Nous y détaillerons également les difficultés que nous avons rencontré et nos différents apprentissages.

La base de données que nous avons exploité a été partagée sur le site de Data Science Kaggle.com et est disponible au lien suivant : [Amazon Products Dataset 2023 \(1.4M Products\) \(kaggle.com\)](https://www.kaggle.com/datasets/amazon-products-dataset-2023). Ce dataset recense plus de 1,4 millions de produits vendus sur la plateforme Amazon US en 2023.

## I / Analyse descriptive et visualisation

1. **Prise en main du DataFrame** : Nous avons lu les données contenues dans les tables suivantes : 'amazon\_products.csv' et 'amazon\_categories.csv'. Puis nous avons décidé de fusionner les 2 tables à l'aide des colonnes : 'category\_id' et 'id'. Puisque les données de ces colonnes étaient identiques dans la table fusionnée, nous avons décidé de supprimer l'une d'entre elles afin de rendre le DataFrame plus lisible.
2. **Découverte des données** : Pour avoir un bref aperçu de notre DataFrame, nous avons employé les commandes : info( ), describe( ) et head( ). Il nous a semblé important de vérifier s'il y avait des valeurs nulle pour les colonnes 'price' et 'stars' car la présence d'un 0 serait trompeuse. En effet, elle ne traduirait pas un prix ou une note nulle mais une valeur manquante dans notre jeu de données. Nous avons donc remplacé les 0 par des NaN.
3. **Visualisation graphique - Question 1** : Nous avons commencé par trier nos variables en fonction des avis dans l'ordre décroissant puis par leur notes. Nous avons ensuite effectué un histogramme pour afficher la note moyenne des 10 articles ayant le plus d'avis.
4. **Visualisation graphique - Question 2** : Par la suite, nous avons décidé d'effectuer un regroupement par catégorie en affichant le prix moyen de ces dernières Puis nous avons choisi de présenter par un histogramme les 15 catégories les plus chères.
5. **Distribution des prix - Question 3** : Nous avons sélectionné les produits Bestseller ayant une note inférieure ou égale à 4. Pour continuer, la réalisation d'une boîte à moustache nous a semblé pertinente pour représenter la distribution des prix. Ainsi le prix médian des produits répondants aux critères de sélection vaut 18,99\$. De plus, 25% de ces biens ont un prix inférieur à 10,99\$ et 75% ont un prix inférieur à 29,99\$. Enfin, nous avons trié ces produits par unité vendues. C'est donc un produit de la catégorie Oral Care Products vérifiant les critères qui a été le plus vendu, à hauteur de 100.000 unités.

6. **Corrélation - Question 4** : Nous avons superposé une matrice de corrélation à une heatmap. Pour ce faire, nous avons sélectionné les variables qui nous semblaient les plus adéquates à savoir : 'price', 'stars', 'reviews', 'isBestSeller', 'boughtInLastMonth'. Effectivement, une corrélation entre 'id' et 'stars' n'aurait aucun sens car ce n'est pas l'identifiant du produit qui aurait un quelconque lien avec sa note.

Parmi les variables citées précédemment, on observe une :

- Corrélation négative (de -0.04) entre stars et price: Quand les prix augmentent, alors la note va avoir tendance à diminuer.
- Corrélation positive (de 0.02) entre stars et reviews : Quand le nombre d'avis sur un produit augmente, alors la note aura aussi tendance à augmenter.
- Corrélation positive (de 0.04) entre stars et BoughtInLastMonth : Plus le nombre d'unités vendues est élevé, plus la note le sera.
- Corrélation positive (de 0.02) entre stars et isBestSeller : Si le produit est Best Seller, alors la note aura tendance à être plus élevée.

Notons tout de même que ces corrélations sont assez faibles.

## II / Recherche de contenu

1. **Fonction 1** : Pour réaliser la fonction permettant à l'utilisateur de rentrer un nom de catégorie, de produit et le type de recherche souhaitée, nous avons créé 3 variables ayant pour valeurs les inputs. Nous créons ensuite un nouveau DataFrame nommé `resultats_prod` qui contient les données correspondant au souhait de l'utilisateur. On passe ensuite au 'filtrage' en fonction du critère indiqué. Enfin, elle affiche les 20 premiers résultats correspondants.
2. **Fonction 2** : Pour réaliser la fonction permettant à l'utilisateur de rentrer un nom de catégorie, une note minimale, un nombre minimal d'unité vendues et le type de recherche souhaitée, nous créons 4 variables ayant pour valeurs les inputs. Puis, nous créons un nouveau DataFrame nommé `resultats` qui contient les données correspondant au souhait de l'utilisateur. Enfin, elle affiche ici aussi les 20 premiers résultats souhaités.
3. **Fonction 3** : Afin de réaliser la fonction de cet exercice, nous avons d'abord créé une fonction `recherche_3_unitaire()` qui permet de filtrer notre data frame selon une note minimale, un budget maximal, un nom de produit et un nom de catégorie. Ensuite notre objectif était de rechercher chaque produit pour chaque catégorie à partir d'une liste de plusieurs produits et une liste de plusieurs catégories. Pour réaliser cela, nous avons dû réaliser une seconde fonction `recherche_3_liste()` qui réalise une double boucle : l'une sur la liste des catégories et l'autre sur la liste des produits. Dans

cette double boucle, nous concaténons les data frames de chaque recherche pour chaque croisement entre un produit et une catégorie.

Enfin, à la fin de la fonction nous réalisons un tri par ordre décroissant sur les variables 'isBestSeller', 'stars', 'reviews'. Notre script « partie\_2\_main.py » permet d'exécuter la fonction recherche\_3\_unitaire() selon les inputs choisies et affiche le data frame filtré dans la console.

### III / Interface graphique

La mise en œuvre de notre interface graphique s'est articulée autour de plusieurs étapes distinctes :

1. **Élaboration de la fonction de recherche des produits** : nous avons développé la fonction recherche\_prod() qui génère un dataframe filtré selon les critères sélectionnés par l'utilisateur. Ces critères comprennent : les noms des produits (nom\_prod), les noms des catégories (nom\_cat), l'intervalle de prix (prix\_min, prix\_max), l'intervalle des notes (note\_min, note\_max), l'intervalle du nombre de ventes (ventes\_min, ventes\_max), la distinction des best-sellers et un choix de tri. Si l'utilisateur entre ces noms en français, ils seront directement traduits en anglais grâce à notre utilisation du module deep\_translator.
2. **Prise en main de Tkinter** : Par la suite, nous nous sommes concentrés sur l'apprentissage et la compréhension du fonctionnement de la bibliothèque Tkinter. Des tutoriels écrits et vidéos nous ont aidé. Grâce à ces ressources, nous avons appris à créer des frames, des widgets (boutons et labels) et à les mettre en place dans la grille (grid) de l'interface.
3. **Affichage du Dataframe** : Nous avons ensuite réalisé l'affichage du data frame complet et non-filtré sur l'interface Tkinter en offrant à l'utilisateur la possibilité de cliquer sur des liens URL pour accéder directement aux pages produits sur amazon.com.
4. **Liaison des Données** : L'étape suivante a consisté à synchroniser le dataframe affiché avec celui filtré, en fonction des données saisies par l'utilisateur via les widgets. Pour ce faire, nous avons mis en place la fonction rechercher(), qui établit cette liaison en créant des variables qui récupèrent les entrées des widgets.
5. **Personnalisation de l'Interface** : Enfin, nous avons personnalisé l'aspect visuel de notre interface en adoptant le code couleur d'Amazon (bleu et orange) et en centrant la fenêtre Tkinter au milieu de l'écran de l'utilisateur.

Voici le résultat obtenu :

Amazon Smart Research

Amazon Smart Research

Recherchez un produit & Cliquez sur un URL

Produits (séparation : virgule)

amplificateur

Prix Min

400

Prix Max

800

Catégories (séparation : virgule)

Audio, Portable

Note Min

4,5

Note Max

Yn

Prix

Ventes Min

Ventes Max

Ordre de tri

Décroissant

Best Seller

☐

Rechercher

Produits trouvés

Naviguez sur amazon.com en cliquant sur les URLs

index	asin	title	imgUrl	productURL	stars	reviews	price	listPrice	category_id	isBestSeller	boughtInLast1	id	category_name
428849	B08DSKPB5L	Bose Music Amplifi	https://m.media-ar	https://www.amaz	4.6	0	699.0	0.0	73	False	0	73	Portable Audio & V
1149688	B07L3CBG1T	New Sonos Wirele	https://m.media-ar	https://www.amaz	4.7	0	699.0	0.0	82	False	0	82	Home Audio & The
1153790	B08FV576L2	MC-84L Stereo Va	https://m.media-ar	https://www.amaz	4.6	0	599.99	0.0	82	False	0	82	Home Audio & The
1150329	B083T84BPQ	Cambridge Audio C	https://m.media-ar	https://www.amaz	4.6	0	599.0	0.0	82	False	0	82	Home Audio & The
1153754	B09WZY394F	Topping PRE90 Pl	https://m.media-ar	https://www.amaz	4.5	0	599.0	0.0	82	False	0	82	Home Audio & The
1153123	B0C59245R5	Topping A70 Pro F	https://m.media-ar	https://www.amaz	5.0	0	499.0	0.0	82	False	0	82	Home Audio & The
1153881	B09P8VGXMX	Klipsch Reference	https://m.media-ar	https://www.amaz	5.0	0	499.0	0.0	82	False	0	82	Home Audio & The
1153086	B08LN49Q8M	XDUOO TA-22 D/A	https://m.media-ar	https://www.amaz	4.7	0	499.0	0.0	82	False	0	82	Home Audio & The
1149987	B07XL4TM3M	Denon PMA-600N	https://m.media-ar	https://www.amaz	4.6	0	449.99	499.0	82	False	0	82	Home Audio & The

Lorsque l'utilisateur clique sur les liens URLs de la ligne sélectionnée en bleu, les fenêtres suivantes s'ouvrent :

- L'image du produit sélectionné (*imgUrl*):



- Et la page web amazon.com du produit sélectionné (*productURL*)



← ↻ 🏠 🔒 https://www.amazon.com/dp/B0BDSKPB5L 🇯🇵 🇬🇧 🇦🇷 🇪🇸 🇮🇹 🇵🇹 🇺🇸 🇬🇧 🇪🇸 🇮🇹 🇵🇹 🇺🇸 🇬🇧 🇪🇸 🇮🇹 🇵🇹 🇺🇸

**amazon** Deliver to **France** Electronics Search Amazon 🔍


☰ All Today's Deals Customer Service Registry Gift Cards Sell

Home Audio & Theater Premium Audio Headphones Home Theater Systems Speakers Wireless Audio Stereo System Components Accessories Deals

Electronics > Portable Audio & Video > Portable Speakers & Docks > Portable Bluetooth Speakers

Roll over image to zoom in



### Bose Music Amplifier – Speaker amp with Bluetooth & Wi-Fi connectivity, Black

[Visit the Bose Store](#)  
4.6 ★★★★★ 65 ratings | 45 answered questions  
50+ bought in past month

**Deal**  
-14% **\$599<sup>00</sup>** (\$299.50 / Item)  
List Price: \$699.00 ⓘ

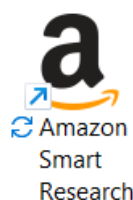
\$178.92 Shipping & Import Fees Deposit to France [Details](#) ▾  
Available at a lower price from [other sellers](#) that may not offer free Prime shipping.

Style: **Amplifier**

<b>Amplifier</b> <b>\$599.00</b> <b>(\$299.50 / Item)</b>	<b>Amplifier + Speakers - White</b> 1 option from \$1,097.00
---	---

<b>Brand</b>	Bose
<b>Model Name</b>	Bose Music Amplifier
<b>Speaker Type</b>	Amplifier
<b>Connectivity Technology</b>	Bluetooth
<b>Recommended Uses For Product</b>	Volume

Enfin, nous avons voulu donner la possibilité de lancer notre programme depuis un Raccourci. Pour ce faire, nous avons créé un fichier .bat comprenant un script qui exécute notre programme main.py. Puis, nous avons créé un Raccourci avec l'icône d'Amazon, issu de ce fichier .bat. En cliquant sur ce Raccourci, notre programme s'exécute :



## IV / Difficultés rencontrées

- En raison du gros volume de données avec lequel nous avons travaillé, nous avons dû réaliser certains de nos exercices avec un échantillon du dataset en phase de test des programmes. Pour ce faire, nous avons utilisé le code `df=df_total.sample(1000)`. Cette méthode nous a permis de raccourcir l'exécution des programmes.

- Pour l’affichage graphique (Partie 1 - Question 1), les données en abscisses étaient illisibles. Il n’était en effet pas possible de distinguer les identifiants des produits car ils se chevauchaient entre eux. Une solution a été de diminuer la police de cet axe à l’aide de ‘fontsize’ qui a permis de rendre le tout plus clair.
- Pour étudier la corrélation entre la note d’un produit et les autres critères (Partie 1 – Question 4), il nous semblait pertinent de réaliser une régression linéaire. Mais nous nous sommes heurtés à une difficulté. En effet, puisque certaines valeurs nulles avaient été corrigées en NaN, la régression ne fonctionnait pas, elle ne pouvait pas traiter de telles valeurs. Une possibilité aurait été de supprimer les lignes contenant des NaN, mais leur trop grand nombre dans la base aurait entraîné une perte d’information importante (environ 10% de l’échantillon). Nous nous sommes donc concentrés sur la matrice de corrélation.
- Nous avons fait face à un autre problème pour l’étude de la corrélation (Partie 1 – Question 4). Nous voulions réaliser un ‘pairplot’ afin d’avoir une visualisation graphique des corrélations. Cependant, les résultats obtenus n’ont pas été satisfaisants. En effet, la fonction nous renvoyait des graphiques étonnants et difficilement interprétables.
- Une des difficultés a été de donner la possibilité pour l'utilisateur de rechercher plusieurs noms de produits et de catégories en séparant les mots par des virgules (Exercice 3 - Partie 2 et Partie 3). Pour offrir cette possibilité, nous avons mis en place une boucle qui parcourt l'ensemble des noms de produits et des catégories choisis par l'utilisateur afin de créer un dataframe final compilant toutes les lignes pertinentes.
- Lors de l’affichage des liens sur Tkinter, nous n’avons pas réussi à afficher les liens URLs en bleu et soulignés. Bien qu’il soit facile de changer la police d’une ligne entière, ajuster la police d’une case spécifique est plus complexe.
- Une autre difficulté a été la gestion des cases vides dans les options entrées par l'utilisateur. Pour remédier à cela, nous avons introduit des conditions if afin de ne convertir en float que les cases non-vides.
- Comprendre le fonctionnement du système de grille (grid) des frames de Tkinter a également été un défi. Cette compréhension était cruciale pour garantir que les boutons ne soient pas désordonnés chaque fois qu’une modification était apportée.



## V / Apprentissage via le projet

### LEBRETON Louis :

- **Rafraîchissement des connaissances sur Tkinter** : ce projet m'a permis de réutiliser cette bibliothèque et de réactualiser mes compétences vis-à-vis de cette interface graphique.
- **Maîtrise de la fonction de clic de la souris** : J'ai également appris à exploiter la fonctionnalité de clic de la souris, ce qui s'est avéré essentiel pour l'interaction utilisateur dans l'interface graphique (afin d'ouvrir les pages web amazon.com).
- **Optimisation & gestion du temps** : J'ai adopté la stratégie du 80/20 dans la priorisation de mes tâches. Cette méthode préconise de se concentrer sur les 80% des tâches les plus cruciales et les plus simples à exécuter plutôt que de s'éparpiller sur les 20% restants, souvent plus complexes et moins impactantes. Cette approche m'a permis d'optimiser mon temps et d'être plus efficace dans la réalisation du projet.

### CHRISTIEN Alexis :

- **Découverte de la documentation de module** : J'ai en effet appris à ajouter de la documentation pour les modules et les fonctions afin de rendre le tout plus accessible pour l'utilisateur. Décrire l'objectif du module et les détails des fonctions sont des éléments primordiaux que j'aurais donc pu découvrir à l'aide de ce projet.
- **Apprentissage des modules** : Ce projet m'a aussi permis d'apprendre à articuler mon travail entre les scripts contenant tout le code et les mains contenant l'import des fonctions. C'est à l'aide de cette nouvelle compétence que j'ai pu organiser les codes de manière plus efficace.

### ABDELJALIL Issame :

- **Approfondissement des notions sur les librairies** : J'ai pu m'entraîner sur la manipulation des dataframes grâce à pandas en respectant les étapes tout au long de la découverte de la base de données. J'ai ainsi appris à vérifier si le nombre de valeurs manquantes n'était pas trop élevé afin de garder de la pertinence dans le dataframe puis à nettoyer les données à l'aide de numpy avant de me lancer dans l'analyse de celui-ci. Pour l'analyse du dataframe, j'ai pu réaliser des représentations graphiques en comparant notamment les différences de visualisation entre ceux réalisés à l'aide de la librairie seaborn par rapport à ceux réalisés sur matplotlib.

- **Réalisation de modules** : J'ai finalement pu apprendre à organiser notre travail en utilisant des modules afin de faciliter la compréhension du projet. Ainsi, j'ai appris à créer des scripts distincts avec pour chaque partie un script 'main' permettant d'exécuter les fonctions contenues dans le script de la partie en question.

**MARZOUK Moustafa :**

- **Fonctions interactives** : À travers la création de fonctions interactives, j'ai eu la chance de consolider mes compétences acquises durant les heures de cours. Les boucles conditionnelles incluses dans ces dernières ont permis aux programmes de prendre des décisions rationnelles, adaptant ses réponses en fonction des préférences spécifiques mentionnées à travers les inputs.  
Les capacités des fonctions de la partie 2, a souligné de plus pour ma part l'impact concret de la programmation sur la résolution de problèmes de l'économie réel, dont je serais confronté dans un avenir proche.
- **Programmation modulaire** : Une stratégie essentielle pour organiser le projet de manière claire et logique a été l'utilisation d'une approche modulaire. En le structurant de cette manière, nous avons réussi à rendre ce dernier plus facilement compréhensible et réutilisable, ce qui a personnellement amélioré ma compréhension de cette approche. En outre, cette méthode a permis une coopération harmonieuse entre les membres de notre groupe.