UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE

Langage de programmation 1 : Evaluation

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 ce que l'on a vu en classe.

Consignes générales

Ce projet sera a 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.

Le livrable devra être composé des éléments suivants:

- L'ensemble de vos **scripts** devront être mis dans un **dossier** nommé :

`sujet_<1 ou 2>_NOMELEVE1_NOMELEVE2_NOMELEVE3_NOMELEVE4`

La qualité et la clarté de vos codes seront pris en considération dans la note du livrable. 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 lisant les scripts.

- Un **rapport** structuré qui doit illustrer votre réflexion pour la réalisation du projet:
    - Explication de votre sujet, de votre démarche, des différents résultats que vous avez obtenus
    - Les instructions pour exécuter votre programme
    - Les éventuelles difficultés que vous avez rencontrées et comment vous avez pu (ou pas) les surmonter (au niveau du groupe mais également au niveau individuel)
    - Prolongements et applications possibles
    - Qu'avez-vous appris ? (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 ~~lundi 21 novembre~~ dimanche 27 novembre 23:59 dernier délai. **Tout retard sera sanctionné**.

La soutenance devra reprendre brièvement les points principaux de votre réalisation et décrire 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 22 novembre~~ 29 novembre 2022). Une démo du programme pourra éventuellement être demandée pendant l'oral.

Tout plagiat sera sanctionné par la note de 0 pour le module entier.

Pour les deux sujets, toute prise d'initiative supplémentaire sera prise en compte et valorisée.

Sujet 1 : Analyse de données Airbnb

L'objectif de ce sujet est d'exploiter des données extraites de la plateforme Airbnb pour la ville de Seattle. Les données comportent des annonces mises sur la plateforme en 2016 principalement (quelques unes peuvent dater de début 2017).

Pour cela, vous avez à votre disposition des données provenant du site Kaggle. Vous pouvez consulter la page de présentation de chacun des fichiers pour avoir plus d'informations, mais aussi pour pouvoir télécharger les données : https://www.kaggle.com/datasets/airbnb/seattle?select=listings.csv

Il y a 3 fichiers de données mais seulement les fichiers suivants seront utiles :

- **calendar.csv**

Libellé de la variable	Définition
listing_id	Identifiant de l'annonce
date	Date
available	Variable binaire indiquant si la location est disponible
price	Prix pour cette nuit

- **listings.csv** (il y a 92 colonnes mais sont présentées ci-dessous uniquement les 10 premières, toutes ne seront pas utilisées)

Libellé de la variable	Définition
Id	Identifiant de l'annonce
listing_url	URL de l'annonce
scrape_id	Identifiant du scrapping (procédé qui a permis l'extraction des données de la plateforme Airbnb)
last_scraped	Date de l'extraction des données
name	Titre de l'annonce
summary	Résumé de la description du logement
space	Description du logement
description	Description de l'annonce
experiences_offered	Variable inutile car constante
neighborhood_overview	Description du quartier

Partie 1 : Analyse descriptive des bases et visualisation:

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

- Quel est le prix moyen des locations par quartier ? Distinguer selon le type de logement (chambre, logement entier, autre). Représenter graphiquement les résultats.
- Quels sont les mois de l'année où il y a le plus de logements disponibles ? Le moins ?
- Quels sont les critères qui semblent influer le plus sur le prix des locations ?
- Quel est le meilleur mois pour visiter Seattle si on souhaite faire des économies ? Quel est le mois pour lequel les locations sont les plus chères ?

Partie 2 : Moteur de recherche

L'objectif de cette partie est de créer un programme qui permette à un utilisateur d'effectuer une recherche de location selon certains critères: il y a quasiment 100 variables dans la base de données donc il ne sera bien évidemment pas possible de laisser l'utilisateur choisir 100 critères. Votre programme devra proposer à minima les options suivantes (vous pouvez ajouter d'autres critères si vous le souhaitez):

- Contraintes au niveau du prix par nuit
- Choix du type de location (appartement entier, chambre…)
- Choix du quartier
- Choix des installations (WiFi, climatisation…)
- Sélection d'une plage de dates pour voir uniquement les annonces de logements qui seront disponibles à cette période

Lorsque les critères choisis par l'utilisateur correspondent à des annonces, les afficher de la moins chère à la plus chère (considérer le prix par nuit dans ce cas). Si aucun résultat de correspond aux critères, il faudra inviter l'utilisateur à changer certains critères.

Il faudra afficher à minima les informations suivantes: Titre de l'annonce, résumé de l'annonce, type de logement, prix par nuit, quartier, nombre de chambres, nom de l'hôte.

Partie 3 : Interface

Une fois le programme de la partie 2 réalisé, il faudra créer une interface afin que l'utilisateur soit guidé pour l'utilisation de votre programme. L'idée est de ne pas devoir écrire des lignes de code pour executer votre moteur de recherche une fois le script entier exécuté. Pour cela, vous devrez utiliser la librairie Python Tkinter[1] 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)

[1] Références pour Tkinter : https://python.doctor/page-tkinter-interface-graphique-python-tutoriel ou https://realpython.com/python-gui-tkinter/

- Affichage des résultats selon ce qui a été demandé dans la partie 2 avec des liens vers les annonces Airbnb et/ou affichage d'images (par exemple l'image principale qui se trouve dans la variable picture_url)

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.*

Partie 4 : Représentation graphique de données cartographiques

Cette partie est indépendante des 3 autres et ne devra pas figurer dans l'interface, vous pouvez mettre l'ensemble du code de cette partie dans un notebook.

L'objectif de cette partie est d'utiliser les librairies Plotly[2] et/ou GeoPandas[3] afin de représenter sur une carte de Seattle la localisation des différentes locations. Si vous rencontrez des soucis de lisibilité des graphiques, vous pouvez choisir de filtrer la base de données et de ne représenter que les annonces répondant à certains critères, en prenant bien soin d'indiquer quel critères vous avez choisi.

Sujet 2 : Top 1000 films et séries sur la plateforme Internet Movie DataBase

Pour cela, vous avez à votre disposition des données provenant du site Kaggle, vous pouvez consulter la page de présentation de chacun des fichiers pour avoir plus d'informations, mais aussi pour pouvoir télécharger les données : https://www.kaggle.com/datasets/ramjasmaurya/top-250s-in-imdb

Il y a 5 fichiers de données disponibles sur le site ne considérer dans le cadre du projet uniquement les bases suivantes:

- **imdb (1000 movies) in june 2022.csv**

Libellé de la variable	Définition
ranking	Classement du film
movie name	Nom du film
year	Année de sortie
certificate	Catégorie d'age (rating)
runtime	Durée
genre	Genre
RATING	Note

[2] Référence pour Plotly : https://plotly.com/python/maps/

[3] Références pour GeoPandas : https://geopandas.org/en/stable/index.html ou https://towardsdatascience.com/geopandas-101-plot-any-data-with-a-latitude-and-longitude-on-a-map-98e01944b972 ou https://geopandas.org/en/stable/docs/reference/api/geopandas.GeoDataFrame.to_crs.html ou https://data.seattle.gov/dataset/City-of-Seattle-Shoreline/vcx7-uj97

Libellé de la variable	Définition
metascore	Score « weighted average of many reviews coming from reputed critics » (source: https://www.freecodecamp.org/news/whose-reviews-should-you-trust-imdb-rotten-tomatoes-metacritic-or-fandango-7d1010c6cf19/)
DETAIL ABOUT MOVIE	Synopsis
DIRECTOR	Réalisateur
ACTOR 1	Acteur principal 1
ACTOR 2	Acteur principal 2
ACTOR 3	Acteur principal 3
ACTOR 4	Acteur principal 4
votes	Nombre de votes
GROSS COLLECTION	Revenu

- **imdb (1000 movies) in june 2022.csv**

Libellé de la variable	Définition
ranking	Classement du film
series name	Nom de la série
year	Année(s) de diffusion
certificate	Catégorie d'age (rating)
runtime	Durée d'un episode
genre	Genre
RATING	Note
DETAILS	Synopsis
ACTOR 1	Acteur principal 1
ACTOR 2	Acteur principal 2
ACTOR 3	Acteur principal 3
ACTOR 4	Acteur principal 4
VOTES	Nombre de votes

Partie 1 : Analyse descriptive des bases et visualisation:

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

- Représenter graphiquement l'évolution du nombre de films par an dans le classement.

- Quelle est la note moyenne par genre ? Faire un graphique
- Existe-t-il un lien entre la note globale des films ou séries et les autres critères présents dans les données ?
- Quel est(sont) le(s) réalisateurs(s) ayant le plus de films apparaissant dans le Top 1000 films ? Combien en a(ont) il(s) ?
- Répondre à la même question que précédemment mais en considérant les acteurs.

Partie 2 : Moteur de recherche

Vous devrez ensuite construire un programme qui aura a minima les 2 fonctions suivantes (Vous pouvez rajouter d'autres fonctionnalités si vous le souhaitez):

1. Recherche d'information:
   - L'utilisateur saisit le nom d'un film ou d'une série: Le programme doit lui retourner (s'il est présent dans la base de données) son classement IMDB, le nombre de notes, l'année de sortie du film, la catégorie d'âge, les acteurs principaux, le genre ainsi que le synopsis
   - L'utilisateur saisit le nom d'un acteur: Le programme doit lui retourner le nom des films dans un premier temps, puis des séries dans lesquels cet acteur a joué (suivi de quelques informations sommaires sur les films). S'il y a plusieurs résultats, les afficher du mieux noté au moins bien noté. Si plusieurs films (ou séries) ont la même note, afficher en premier celui qui a reçu le plus de votes
   - L'utilisateur saisit le nom d'un réalisateur: Le programme doit retourner les films et les séries agrégées par année (du plus récent au plus ancien). S'il y a plusieurs films ou séries pour la même année, afficher le revenu total par année pour les films de ce réalisateur
   - L'utilisateur saisit une année et un genre: top 3 des meilleures films et top 3 des meilleures séries
2. Recommendation de contenu:
   - Pour cette partie, le programme devra aider un utilisateur à choisir un film ou une série à regarder en fonction de ses choix en lui demandant des informations sur ses préférences: série ou film, genre, catégorie d'age, durée d'épisode (pour une série) ou du film et éventuellement le nom d'un ou plusieurs acteurs ou de réalisateur. Le programme doit laisser la possibilité à l'utilisateur de ne pas sélectionner toutes les options. *Par exemple si je demande un film d'action sans contrainte de catégorie d'age qui dure moins de 3h avec Tom Holland dans le casting, le programme devrait me suggérer Spider-Man: No Way Home*
   - S'il y a plusieurs résultats correspondant aux critères de l'utilisateur, les ordonner par note croissante ou par durée (choisissez le critère que vous souhaitez)

Partie 3 : Interface

Le but de cette interface sera de reprendre le fonctionnement du programme de la partie 2 afin de guider l'utilisateur pour l'utiliser. L'idée est de ne pas devoir écrire des lignes de code pour executer votre programme une fois le script entier exécuté. Pour cela, vous devrez utiliser la librairie Python Tkinter[4] pour réaliser une interface graphique. Cette interface devra comporter les éléments suivants:

- Choix de fonctionnalité: « Recherche d'informations » ou « Recommendation de contenu »
- Possibilité pour l'utilisateur de saisir les différents critères (selon ce qui est demandé et qui sera fait à la partie 2)
- Affichage des résultats conformément aux éléments demandés à la partie 2.

Idée: Vous pouvez améliorer l'interface en y intégrant des liens vers les pages Wikipedia, IMDB ou ce qui vous semble pertinent.

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.*

[4] Références pour Tkinter : https://python.doctor/page-tkinter-interface-graphique-python-tutoriel ou https://realpython.com/python-gui-tkinter/