Exercice partie 1 - Analysez des données de systèmes éducatifs

Conseils mentor

L'étudiant peut potentiellement réaliser cet exercice en quelques heures, sans même une session de mentorat.

Cependant, s'il a des difficultés pendant l'exercice et que vous avez une session de mentorat, demandez-lui de vous parler de :

1. son avancement ;
2. ses difficultés ;
3. sa compréhension de l'exercice.

Vous pourrez ainsi identifier ses problèmes, ses lacunes, ses processus ou méthodologies erronés.

- **Compréhension des attendus**

Assurez-vous que l'étudiant puisse répondre à ces questions : Que contiennent les livrables ? À quoi ressembleront-ils ? Quel est le niveau de précision attendu ?

- **Méthodologie**

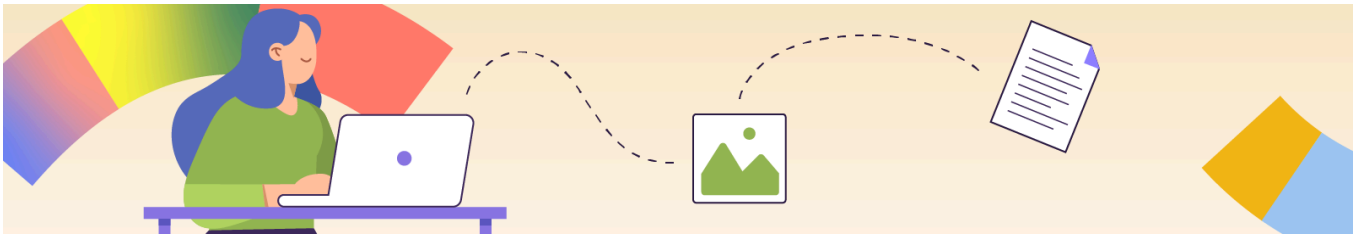
Si l'étudiant a du mal a prendre en main Python et Pandas et à comprendre comment lire des erreurs de code, montrez lui comment vous vous y prenez pour débugger du code, ou travailler avec un package nouveau pour vous (StackOverflow, GeeksforGeeks, lecture de la documentation des packages pour comprendre les entrées et sorties des fonctions etc.). Essayez de planter ces graines qui lui permettront d'être le plus autonome possible.

- **Points de vigilance**

Si l'étudiant est très à l'aise car il a déjà une expérience en programmation, concentrez vous sur la qualité de l'analyse statistique. Si l'étudiant est déjà à l'aise en analyse statistique, concentrez-vous sur la qualité du code.

Voici les documents qui vous aideront à faire un bilan avec l'étudiant :

- Sa fiche d'autoévaluation
- Elle permet de faire le point sur ce que l'étudiant a appris. Il la complète seul et en discute avec vous.



Qu'allez-vous faire et comment ?

Pour rappel, avant de démarrer votre travail sur ce projet, nous vous conseillons de suivre attentivement les cours.

- lire tout le projet qui inclut 3 exercices liés et leurs documents ;
- prendre des notes sur ce que vous avez compris ;
- réaliser une fiche de synthèse de ce que vous en avez retenu ;
  - Essayez de reformuler ce dont vous vous souvenez sans consulter le cours dans un premier temps pour renforcer vos connaissances.
  - Nous vous conseillons de créer une page Notion pour ce projet, ou de créer un document que vous alimenterez au fur et à mesure.
- préparer une liste de questions pour votre première session de mentorat notamment notez tout ce qui vous semble peu clair.

Prêt à résoudre l'exercice ?

Dans cet exercice, vous êtes Data Scientist dans une **start-up de la EdTech**, nommée ***academy***, qui propose des contenus de formation en ligne pour un public de niveau lycée et université.

Mark, votre manager, vous a convié à une réunion pour vous présenter le projet d'**expansion à l'international** de l'entreprise. Il vous confie **une première mission d'analyse exploratoire**, pour déterminer si les données sur l'éducation de la banque mondiale permettent d'enrichir la réflexion autour du projet d'expansion

Mark aimerait explorer les pays avec un fort potentiel de clients pour les services de ***academy***, et voir comment ce potentiel pourrait évoluer.

Vous allez répondre à la demande de Mark en suivant l'ensemble des exercices entièrement guidés. A la fin des exercices, vous aurez réussi à déterminer si ces jeux de données peuvent fournir des insights guidant ***academy*** à décider dans **quels pays s'implanter**.

Ce premier exercice vous fait mener une première analyse en surface des différents jeux de données.

Suivez les étapes ci-dessous.

Étapes

Étape 1 - Chargez les données dans votre Notebook

Voici le **lien** vers le jeu de données issu du site de la Banque Mondiale.

Prérequis

- Avoir suivi les instructions de la section Cours précédente (en fonction de votre niveau).
- Avoir installé **Python** et **JupyterLab** (ou **Jupyter Notebook** le cas échéant) sur mon ordinateur via l'outil pip ou anaconda.

Nous vous recommandons d'installer la version la plus récente de Python.

Résultats attendus

- Avoir chargé les données.
- Visualiser dans un Jupyter Notebook les premières lignes des cinq fichiers en utilisant Pandas (présenté dans le cours ci-dessus).

En cas de difficultés de compatibilité entre votre système/votre ordinateur et l'outil d'installation, vous pouvez réaliser l'exercice via Google Colab.

- Prenez le temps de comprendre ce que signifie une ligne dans chaque fichier de données.
- Tirez parti au maximum des différentes options du Jupyter Notebook :
  - Assurez-vous que chaque case correspond à une tâche spécifique.
  - N'hésitez pas à ajouter des commentaires et des markdowns pour faciliter la navigation.

Ressource

- Google Colab

Étape 2 - Collectez des informations basiques sur chaque jeu de données

Résultat attendu

- Code et markdown dans un Jupyter Notebook, permettant de reproduire les réponses aux instructions ci-dessous.

Quand nous vous demanderons du code dans les étapes intermédiaires de cet exercice, gardez à l'esprit que nous n'attendons pas un rendu d'expert. Il faut simplement que votre code soit fonctionnel et obéisse à certaines bonnes pratiques, mais il ne doit pas nécessairement être optimisé.

Instructions

- Pour chaque fichier, suivez ces instructions :
  - Définissez ce que représente une ligne.
    - une ligne = un pays ? un indicateur ? une combinaison des deux ? autre chose ?
  - Calculez le nombre de lignes et de colonnes.
  - Calculez le nombre de doublons dans le jeu de données.
    - Supprimez les doublons s'il y en a.
  - Calculez la proportion de valeurs manquantes par colonne.
    - Supprimez les colonnes inutilisables.
  - Pour les colonnes numériques : calculez les statistiques descriptives basiques en utilisant describe().
  - Pour les colonnes catégorielles : calculez le nombre d'occurrences de chaque valeur possible de la colonne.

Recommandations

- Réutilisez le plus possible les méthodes déjà implémentées dans Pandas :
  - head()
  - shape
  - unique()
  - duplicated()
  - drop_duplicates()
  - value_counts()
  - info()
  - isnull() etc.
- Traitez chaque fichier de données l'un après l'autre.
- Si vous rencontrez des erreurs de code que vous ne comprenez pas, copiez le message d'erreur et collez-le dans votre moteur de recherche.
  - Vous aurez très probablement comme résultats des pages du forum StackOverflow, où quelqu'un aura déjà posé la question.
  - Une majorité écrasante des erreurs de code ont déjà été rencontrées, publiées sur StackOverflow par d'autres personnes, et résolues par la communauté d'experts qui l'anime.

Nous vous déconseillons à ce stade d'utiliser ChatGPT (ou un équivalent) pour débugger votre code. En effet, vous pourriez obtenir des réponses trompeuses et perdre du temps au final.

Ressources

- Le chapitre du cours "Découvrez les librairies Python pour la Data Science " sur le filtrage des données d'un dataframe.
- Le webinaire associé au projet qui présente pas à pas un **exemple pratique** d'exploration et de nettoyage de données.
- Le repo de la librairie missingno qui permet de simplifier l'analyse et la visualisation des différentes données manquantes d'un dataframe.

Étape 3 - Réalisez votre premier nettoyage

Résultat attendu

- Code permettant de filtrer les faux pays des tables où cela fait sens (Country, Country-Series, FootNote et Data).
- Markdown associé au code pour expliquer l'approche.

Instructions

- Regardez de plus près les lignes du fichier Country pour identifier des faux pays.
- Supprimez les lignes correspondantes du dataframe contenant la donnée Country.
- Utilisez les 2 méthodes suivantes pour supprimer les faux pays des autres dataframes :
  - En stockant les faux pays dans une liste qui sera utilisée pour le filtrage des différents dataframes.
  - En utilisant un inner join entre les pays du dataframe Country nettoyé, et les autres dataframes.

Recommandations

- L'objectif ici est de déterminer si le jeu de données Country contient bien ce qu'il est censé contenir : des informations sur des pays.

Il ne faut jamais partir du principe qu'un dataset est à 100% fiable. Au contraire, il faut systématiquement vérifier si des informations peu fiables sont présentes, et faire le nécessaire pour fiabiliser le dataset.

- Utilisez le filtrage à base de conditions avec Pandas comme expliqué dans le chapitre (voir la section Ressource ci-dessous).
- Pour filtrer des valeurs indésirables dans un dataframe, utilisez la formule :
  - df[~df[colonne].isin(liste_mauvaises_valeurs)]
  - L'utilisation du caractère ~ permet d'inverser la condition spécifiée à Pandas lors du filtrage.

Utilisez des markdowns pour bien organiser votre notebook au fur et à mesure des différents travaux réalisés. Sans markdown avec des sections claires que l'on peut cacher au besoin, votre notebook peut facilement devenir trop long et difficile à utiliser.

Points de vigilance

- Évitez l'utilisation d'iloc pour supprimer des lignes ou des colonnes sur la base d'une position d'index.
  - En effet, des traitements de données peuvent changer les index associés à une ligne ou une colonne (par exemple en triant le dataframe selon une colonne, ou en supprimant une colonne).
  - Si vous spécifiez en dur l'index d'une ligne et d'une colonne, et que vous avez transformé votre dataframe entre temps, vous supprimerez la mauvaise ligne.
  - Cette mauvaise pratique s'appelle le hardcoding, et il faut systématiquement l'éviter.

- quelles années conserver ou non ;
- comment gérer les valeurs manquantes, etc.) et
- comment les organiser.

Ressource

- Chapitre Filtrez les données du data frame du cours "Découvrez les librairies Python pour la Data Science".

Étape 4 - Vérifiez votre travail et faites le point avec votre mentor

Pour vérifier que vous n'avez rien oublié dans la réalisation de votre exercice, téléchargez et complétez la **partie 1** de la fiche d'autoévaluation.

Parlez-en avec votre mentor durant votre dernière session de mentorat

Avez-vous une suggestion pour nous ?

Précédent Suivant