Pandas : Bibliothèque open-source Python pour la Data Science

Pandas est conçue pour la manipulation et l’analyse de données. Voici les étaoes : le nettoyage, la transformation, l’analyse, la modélisation, la visualisation et le reporting.

**Installation :**

sur Windows **: py -m pip install pandas**

**Les données** : elles sont sous forme de tableaux/matrices/séries temporelles. On les appelle “[DataFrames](https://datascientest.com/quest-ce-quun-dataframe)” : des tableaux de données en deux dimensions. Les données stockées dans un DataFrame peuvent être des nombres ou des caractères. Pandas permet d’importer et d’exporter les données dans différents formats comme CSV ou JSON, SQL ou Excel.

**Fonctionnalités** : rendre des dataframes exploitables, nettoyer les données, compenser les données manquantes. Les colonnes peuvent être facilement insérées ou supprimées au sein des DataFrames.

Opération du type « split-apply-combine » :

1- diviser les données en groupes en fonction de certains critères

2- appliquer une fonction à chaque groupe indépendamment.

3- combiner les résultats dans une structure de données.

En revanche, Pandas n’est pas forcément adéquat pour travailler avec des formats de données incompatibles comme des images, des fichiers audio ou certaines données textuelles. La structure de ces types de données n’est pas adaptée pour un usage avec Pandas. Il est donc important de tenir compte du type de données à traiter avant de choisir un outil.

**Code :**

import numpy as np

import pandas as pd

**LIRE/AFFICHER :**

Lire un tableur : df = pd.read\_csv("fichier.csv")

Afficher les 5 premières lignes : df.head()

Afficher la taille (colonne x ligne) de df : print(df.shape)

Affiche le nom des colonnes : print(df.columns)

Affiche qlqs info a propos des données : print(df.info())

Changer le type de variable d’une colonne : df[“nom\_colonne1”]=df[“nom\_colonne1].astype(“int64”)

Afficher les caractéristiques statistiques de base de chaque caractéristique numérique (types int64 et float64) : nombre de valeurs non nulles, moyenne, écart-type, plage, médiane, quartiles 0,25 et 0,75 : df.describe()

Pour afficher les caractéristiques statistiques de colonne non numérique : df.describe(include=["object", "bool"])

Calculer les valeurs totales par colonnes : df["Churn"].value\_counts()

Possibilités de normaliser les données avec : df["Churn"].value\_counts(normalize=True)

**TRIER :**

Trier selon 1 colonne : df.sort\_values(by="Total day charge", ascending=False).head()

Trier selon plusieurs colonnes : df.sort\_values(by=["Churn", "Total day charge"], ascending=[True, False]).head()

garder une seule colonne : DataFrame['Name']

faire la moyenne : df[“colonne1”].mean()

etc…

Voir toutes les méthodes sur [Topic 1. Exploratory Data Analysis with Pandas | Kaggle](https://www.kaggle.com/code/kashnitsky/topic-1-exploratory-data-analysis-with-pandas)