

SLIDE 1 – Page de garde

À dire :

Bonjour Monsieur le Professeur, bonjour à tous.

Nous allons vous présenter notre projet Python intitulé **API de Data Processing**.

Ce projet a été réalisé dans le cadre du Master 1 Data Science et Intelligence Artificielle à l’Institut Supérieur d’Informatique.

SLIDE 2 – Contexte

À dire :

En Data Science, le traitement des données est une étape indispensable avant toute analyse ou modélisation.

Cependant, les données brutes contiennent souvent des valeurs manquantes, des valeurs aberrantes et des doublons.

Ces tâches de nettoyage sont répétitives et prennent beaucoup de temps, surtout lorsqu’elles sont faites manuellement.

SLIDE 3 – Problématique

Partant de ce constat, nous nous sommes posé la question suivante :

comment automatiser efficacement le nettoyage des données tout en proposant une solution simple et accessible aux utilisateurs ?

SLIDE 4 – Objectif général

L’objectif général de notre projet est de mettre en place une **API** capable d’automatiser le nettoyage et la normalisation des données,

afin de faciliter leur préparation pour l’analyse ou le machine learning.

SLIDE 5 – Objectifs spécifiques

Plus précisément, notre solution permet de détecter et traiter les valeurs manquantes, d’identifier et corriger les valeurs aberrantes, de supprimer les doublons, de normaliser les données numériques, et enfin de proposer une interface graphique simple pour l’utilisateur.

SLIDE 6 – Description du projet

Concrètement, le projet consiste à développer une **API avec Flask**. L'utilisateur téléverse un fichier de données via une interface web, le fichier est ensuite traité automatiquement par l'API, et un fichier nettoyé est généré et mis à disposition pour le téléchargement.

SLIDE 7 – Fonctionnalités de l'API

Notre API offre plusieurs fonctionnalités principales : l'importation de fichiers, le nettoyage automatique des données, la normalisation des variables numériques, et enfin l'exportation du fichier final prêt à être utilisé.

SLIDE 8 – Formats et technologies

L'outil supporte plusieurs formats de fichiers tels que le CSV, Excel, JSON et XML. Sur le plan technique, le backend est développé en Python avec Flask, tandis que le frontend utilise HTML, CSS et JavaScript. Nous avons également utilisé les librairies Pandas, NumPy et Scikit-learn pour le traitement des données.

SLIDE 9 – Architecture du système

L'architecture de notre solution est simple et modulaire. L'interface web communique avec l'API Flask via des requêtes HTTP. L'API appelle ensuite le module de nettoyage des données, puis génère le fichier propre qui est retourné à l'utilisateur.

SLIDE 10 – Conclusion

Pour conclure, ce projet permet d'automatiser une étape essentielle du cycle de la Data Science.

Il offre un gain de temps considérable et améliore la qualité des données. Des améliorations futures pourraient inclure davantage d'options de nettoyage ou l'ajout d'une base de données.

Merci pour votre attention, nous sommes disponibles pour répondre à vos questions.

