

Projet de fin de module

Data Visualization – Master 2 BDIA (Big Data & Intelligence Artificielle)

1. Objectif pédagogique

Ce projet constitue **l'évaluation principale du module de Data Visualization**.

Il vise à évaluer votre capacité à :

- Formuler et comprendre une **problématique métier ou analytique**
- Choisir et concevoir des **visualisations pertinentes**, lisibles et efficaces
- Appliquer rigoureusement les **bonnes pratiques de visualisation** vues en cours
- Automatiser le raisonnement de visualisation à l'aide de **LLMs**
- Concevoir une **application web fonctionnelle**, proprement développée et déployée
- Travailler en **collaboration logicielle** en utilisant des standards professionnels

2. Organisation du projet

- Le projet est à réaliser **en binôme obligatoire**
- Les binômes doivent être déclarés **au plus tard demain** via [ce Google Sheet](#)

3. Sujet du projet

Vous devez développer une **application web intelligente de data visualisation** permettant de répondre automatiquement à une problématique donnée.

Fonctionnalités attendues

Votre application doit :

1. Prendre en entrée :

- Une **problématique textuelle** (ex. : *"Quels facteurs influencent le prix des logements à Paris ?"*)
- Un **dataset tabulaire** (CSV ou équivalent) lié à cette problématique

2. Générer automatiquement (via un ou plusieurs LLMs) :

- **Trois propositions de visualisations différentes**, chacune :
 - Justifiée par rapport à la problématique

- Conforme aux bonnes pratiques vues en cours
(choix du type de graphique, lisibilité, absence de chartjunk, data-ink ratio optimal, etc.)

3. Interaction utilisateur :

- L'utilisateur sélectionne **une des trois propositions**
- L'application génère alors automatiquement :
 - La **visualisation finale**
 - Correctement formatée (titres, axes, légendes, couleurs, annotations si pertinentes)

4. Export :

- La visualisation finale doit être **téléchargeable au format PNG**

4. Contraintes techniques

Intelligence artificielle

- Vous devez utiliser **un ou plusieurs LLMs** pour automatiser :
 - L'analyse de la problématique
 - Le choix des visualisations
 - La génération (ou le pilotage de la génération) des graphiques
- L'usage de **scaffoldings LLM** est attendu (raisonnement en étapes)

Déploiement

Deux options possibles :

Option A – Python only

- Utilisation d'une librairie comme :
 - [Streamlit](#) ou [Gradio](#)
- Déploiement public sur **Hugging Face Spaces**

Option B – Web complet

- Frontend + Backend
- Déploiement sur un **domaine personnel**
- Stack libre (React, FastAPI, etc.)

L'application doit être **accessible publiquement** au moment du rendu.

5. Code & bonnes pratiques de développement

Vous devez fournir un **repository GitHub public**, respectant les standards professionnels vus en cours.

Structure et qualité du code

Le repository doit contenir :

- Une architecture claire de type **src/ layout**
- Un gestionnaire de dépendances :
 - **uv ou poetry**
- Un **README.md** clair comprenant :
 - Description du projet
 - Instructions d'installation
 - Instructions de lancement
 - Lien vers l'application déployée
- Une **LICENSE**
- Des **tests** (même simples)
- Une documentation minimale

Collaboration

- Le repository doit clairement montrer une **collaboration réelle** :
 - Pull requests entre les deux membres
 - Commits distincts

6. Livrables

À rendre à la date qui sera communiquée :

1. Lien vers l'**application déployée**
2. Lien vers le **repository GitHub**
3. Une **note comprenant 3 exemples** de problématiques / datasets et les résultats de votre application sur ceux-ci.
4. Une **vidéo courte de 3 min** montrant l'utilisation de votre application en live

Deadline : Samedi 7 février 2026 à minuit