# **AI (ML)&BIG DATA Assignment**

**Instructor**: Victor Muv  
**GitHub Repo**: <https://github.com/Lecturing-Asseignments>

**Tools Required**

* **Python + Jupyter Notebook**: Install via [Anaconda](https://www.anaconda.com/download" \t "_blank).
* **Git**: [Download here](https://git-scm.com/downloads).
* **GitHub Account**: Sign up at [github.com](https://github.com/).

**Objective**: Complete a full data science pipeline (Data Cleaning → EDA → ML → UI) using Python and GitHub.

**Deadline**: 14 June 2025

**Submission**: Via GitHub (details below).

**1. Setup Instructions**

**A. Create a GitHub Account**

1. Sign up at [github.com](https://github.com/).
2. Install Git:
   * Windows: Download [Git Bash](https://git-scm.com/downloads).
   * Mac/Linux: Run git --version (install if missing).

**B. Configure Git**

Run these commands once:

git config --global user.name "Your Name"

git config --global user.email [your@email.com](mailto:your@email.com)

**2. Assignment Tasks**

**Task 1: Data Cleaning**

* **Dataset**: Choose from [Titanic](https://www.kaggle.com/c/titanic/data), [Iris](https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_iris.html), or [Housing Prices](https://www.kaggle.com/c/house-prices-advanced-regression-techniques).
* **Steps**:
  + Load data with Pandas.
  + Handle missing values (dropna() or fillna()).
  + Remove duplicates.
  + Save cleaned data as cleaned\_data.csv.

**Task 2: Exploratory Data Analysis (EDA)**

* **Tools**: Use Matplotlib/Seaborn.
* **Deliverables**:
  + Summary statistics (df.describe()).
  + At least 3 visualizations (e.g., histograms, heatmaps).
  + Pandas Profiling report (optional).

**Task 3: Machine Learning Model**

* **Requirements**:
  + Train a model (e.g., RandomForestClassifier).
  + Split data into train/test sets (train\_test\_split).
  + Evaluate accuracy (or other metrics).

**Task 4: Create a UI**

* **Options**:
  + **Streamlit**:

import streamlit as st

st.title("My Model")

st.write("Prediction:", model.predict([input]))

**Gradio** (simpler)

import gradio as gr

gr.Interface(fn=predict, inputs="number", outputs="label").launch()

**3. Submission Guidelines**

**A. Repository Setup**

1. **Fork** this repo: [Your-Repo-Link] (if provided).  
   OR  
   **Create a new repo** named [YourName]-ml-assignment.
2. **Clone your repo locally**:

git clone [https://github.com/[YourUsername]/[RepoName].git](https://github.com/%5bYourUsername%5d/%5bRepoName%5d.git)

**B. Required Files**

* notebook.ipynb (Jupyter Notebook with all tasks).
* requirements.txt (Python dependencies).
* README.md (project description).

git add .

git commit -m "Submission by [Your Name]"

git push origin main

**D. Submit Pull Request (if forked)**

1. Go to your repo on GitHub → **Pull Requests** → **New PR**.
2. Add a title: [Your Name] - ML Assignment.

**Grading Rubric**

**Total Points: 100**

| **Category** | **Criteria** | **Points** | **How to Earn Full Credit** |
| --- | --- | --- | --- |
| **1. Data Cleaning** | - Missing values handled properly. - Duplicates removed. - Data saved in clean format. | 20 | Provide code + comments explaining steps. |
| **2. EDA** | - At least 3 meaningful visualizations. - Summary statistics included. - Insights clearly stated. | 25 | Visualizations are labeled and interpretable. |
| **3. ML Model** | - Correct train/test split. - Model trained and evaluated. - Accuracy/report included. | 30 | Use sklearn.metrics for evaluation. |
| **4. UI** | - Functional interface (Streamlit/Gradio). - Connects to the model. - Input/output working. | 15 | Test the UI locally before submission. |
| **5. GitHub Submission** | - Repo contains all files. - README with project description. - Code is well-organized. | 10 | Follow naming conventions (e.g., notebook.ipynb). |

**5. Resources**

* Python Cheatsheet: [DataCamp](https://www.datacamp.com/cheat-sheet/python-for-data-science" \t "_blank)
* Git Tutorial: [GitHub Guides](https://guides.github.com/activities/hello-world/)