📓 Assignment 1: air quality Pollution Forecasting (Regression)

🧠 **Difficulty**: Advanced | 🕒 **Time**: 4–5 hours
🧰 **Tools**: Pandas, NumPy, Scikit-learn, Seaborn, Matplotlib, OOP, Streamlit or MLflow
📁 **Dataset**: Air Quality - UCI Machine Learning Repository

🔍 Problem Statement

You are building a predictive model for an environmental agency that can forecast **air pollution concentration** based on weather data. The model will be used in a web application to inform citizens about air quality.

☑️ Assignment Objectives

1. **Build an end-to-end regression pipeline** using object-oriented design.
2. Include **feature engineering, preprocessing, model training**, and **evaluation**.
3. Deploy the model using **Streamlit** or **track experiments** using **MLflow**.

🧱 Task Breakdown

Task 1: Data Wrangling & Preprocessing

- Implement a class `PollutionDataHandler` with:
  - `load_data()` to load the CSV
  - `clean_data()` to handle missing values
  - `engineer_features()` to:
    - Merge date/time columns into one `datetime` column
    - Extract `hour`, `month`, `day_of_week`
  - `scale_features()` for normalization

Task 2: Model Training & Evaluation

- Create class `PollutionPredictor` with:
  - `train_model()`: Use RandomForestRegressor and Linear Regression
  - `evaluate_model()`: Print MAE, RMSE, R^2
  - `cross_validate()` function for robust evaluation
- Visualizations:
  - Feature importance (barplot)
  - Residual plot

Task 3: Pipeline & Deployment

Choose one of the following:

✅ **Option A – Streamlit UI**

- Allow users to input features via form sliders (temperature, wind, etc.)
- Output PM2.5 prediction
- Add charts: residuals, feature importance

✅ **Option B – MLflow Integration**

- Use MLflow for:
  - Logging hyperparameters
  - Logging metrics
  - Saving models
- Register the best model and use `mlflow.pyfunc.load_model()` for inference

📓 Assignment 2: Banknote Authentication (Classification)

🧠 **Difficulty**: Advanced | 🕒 **Time**: 3–4 hours
🧰 **Tools**: Pandas, NumPy, Scikit-learn, Seaborn, Matplotlib, OOP, MLflow or Streamlit
📁 **Dataset**: Banknote Authentication - UCI Machine Learning Repository

🔍 Problem Statement

As a consultant for a security firm, you're tasked with building a classification model that detects **fake banknotes** using statistical image features. This model should be available either as a **web interface** or integrated in a **deployment pipeline**.

☑ Assignment Objectives

1. **Object-oriented classification system** using scikit-learn
2. Apply **EDA**, **feature selection**, and **model evaluation**
3. Use **Streamlit or MLflow** for deployment/tracking

🧱 Task Breakdown

Task 1: Data Handling & Analysis

- Build class `BanknoteHandler`:
  - `load_data()` to read and label columns
  - `preprocess()` to standardize data
  - `plot_distribution()` to visualize fake vs real banknotes

Task 2: Model Classifier & Evaluation

- Build class `BanknoteModel`:
  - `train_model()` using Logistic Regression, RandomForest
  - `evaluate_model()` to print accuracy, F1-score, confusion matrix
  - `predict_note(features)` to predict a new sample
- Visualize:
  - Confusion matrix
  - ROC-AUC Curve
  - Feature importances

Task 3: Full Pipeline & Deployment

✅ **Option A – Streamlit**

- Create a form with sliders for input features (variance, skewness, etc.)
- Display prediction: Real or Fake
- Add performance charts

✅ **Option B – MLflow**

- Log:
  - Models
  - Metrics
  - Parameters
- Use `mlflow models serve` to expose the trained classifier as a REST API