📓 Assignment 1: Time Series Forecasting for Energy Consumption

🧠 **Difficulty**: Advanced | 🕒 **Time**: 4–5 hours
🧰 **Tools**: Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn, Statsmodels, OOP, Streamlit or MLflow
📁 **Dataset**: Individual household electric power consumption

🔍 Problem Statement

You are working as a Data Scientist in an energy management company. Your task is to predict the **next-day total global active power consumption (in kW)** using historical electricity data. The business will use this to **automatically adjust energy supply**.

☑️ Assignment Objectives

1. Forecast future energy consumption using statistical models (ARIMA, SARIMA).
2. Create classes and functions for each pipeline step.
3. Track experiments or deploy forecast UI using Streamlit or MLflow.

🧱 Task Breakdown

Task 1: Data Wrangling & Exploration

- Create a class `EnergyDataProcessor` with:
  - `load_data()` to read and parse datetime column.
  - `resample_data(freq='D')`: daily aggregation.
  - `plot_consumption_trend()` to visualize trends and seasonality.

Task 2: Forecasting Models

- Create a class `TimeSeriesForecaster`:
  - `train_model()` to fit SARIMA or Holt-Winters
  - `forecast_next_days(n)` to predict next n days
  - `evaluate_forecast()` with RMSE, MAE

Task 3: Pipeline & Deployment

✅ **Option A – Streamlit UI**

- Date picker for forecasting horizon
- Display forecast plot and confidence intervals

✅ **Option B – MLflow**

- Track:
  - Model type, parameters
  - Evaluation metrics
- Register the best-performing model

🗂️ Deliverables

- Jupyter notebook or `.py` file with:
  - Class-based architecture
  - Data pipeline + visualizations
- Streamlit app or MLflow experiment
- README with setup instructions

📓 Assignment 2: LSTM-Based Weather Forecasting (Deep Learning)

🧠 **Difficulty**: Expert | 🕒 **Time**: 5–6 hours

🧰 **Tools**: TensorFlow/Keras, Pandas, NumPy, Seaborn, Matplotlib, Scikit-learn, Streamlit

or MLflow

📁 **Dataset:** Daily weather data from the city of Szeged, Hungary

🔍 Problem Statement

A smart agriculture startup needs to **forecast future temperatures** to automate irrigation and optimize crop health. Your role is to create a **deep learning model using LSTM** that learns from historical weather features.

☑️ Assignment Objectives

1. Build an LSTM-based model for time series prediction.
2. Use a modular object-oriented approach (class/functions).
3. Deploy using **Streamlit** for interaction or **MLflow** for model lifecycle.

🧱 Task Breakdown

Task 1: Data Preprocessing

- Create class `WeatherDataLoader`:
  - `load_data()` to load CSV
  - `create_features(target='temp')`: scale data and generate lag features
  - `create_sequences(window_size)` to feed into LSTM

Task 2: LSTM Modeling

- Create class `LSTMForecaster`:
  - `build_model()`: define LSTM model using Keras
  - `train_model()` with early stopping
  - `forecast()` and `evaluate()` with RMSE, MAE

Task 3: Visualization

- Plot training history, predictions vs actual, and loss curve

Task 4: Deployment

✅ **Option A – Streamlit**

- Input: days to forecast + model params
- Output: Forecasted temperature graph, loss plot, slider to adjust window size

✅ **Option B – MLflow**

- Log:
  - Model architecture summary
  - Epochs, loss, validation loss
  - Metrics and parameters

🗂️ Deliverables

- `lstm_forecaster.py` (model class)
- `data_handler.py` (data class)
- Streamlit UI OR MLflow tracking notebook
- Saved `.h5` or `joblib` model
- README.md with instructions