Information Note: Economic Monitoring Dashboard for Renewable Energy

Executive Summary

This document presents a detailed analysis of the project **"Economic Monitoring Dashboard – Renewable Energy in Francophone Africa"**. Its main objective is to provide an interactive tool for the collection, visualization, and predictive analysis of renewable energy data. Built with Streamlit, the application leverages the **World Bank API** to automatically retrieve key indicators for multiple Francophone African countries. Its architecture integrates a **Bayesian multivariate linear regression model with PyMC** for forecasting, alongside a configurable **email alert system** via SMTP. The user interface is dynamic, allowing filtering by country, indicator, and period, and offers interactive visualizations using **Plotly** and **Matplotlib**. The project is designed for extensibility, with possibilities for integrating additional data via web scraping and deployment on cloud platforms.

1. Project Overview

- **Project Name:** Economic Monitoring Dashboard – Renewable Energy in Francophone Africa
- **Objective:** Provide an interactive tool to automate the collection, visualization, and Bayesian analysis of renewable energy data. Data are sourced from the World Bank and cover multiple Francophone African countries.

2. Architecture and Technical Components

The application is structured around several key functional modules, from data acquisition to user presentation.

2.1. Data Collection

- **Source:** The World Bank API is used for automatic retrieval of indicators.
- **Key Indicators:** "Renewable energy share (%)" and "Renewable electricity production (kWh)".
- **Scope:** Multi-country and multi-indicator retrieval over configurable time ranges (default: 2000–2024).
- **Optimization:** The fetch_worldbank_data() function uses **Streamlit caching (@st.cache)** to minimize redundant API calls and improve performance.

2.2. Bayesian Analysis

- **Model:** A Bayesian multivariate linear regression is implemented using **PyMC**.
- **Forecasting:** Users can generate predictions for a configurable number of future years (default: 3 years).
- **Model Validation:** MCMC diagnostics (Markov Chain Monte Carlo) assess model convergence and quality, including trace plots, R-hat tests, and autocorrelation analysis.

2.3. Data Visualization

- **Interactive Charts:** Time series are generated with **Plotly**, allowing interactive exploration by country and indicator.
- **Bayesian Results:** Predictive results, including credibility intervals, are displayed using **Matplotlib**.
- **Summary Tables:** A concise table summarizes observed data and forecasted values.

2.4. Alert System

- **Functionality:** An optional **email alert** notifies users if an indicator exceeds a predefined threshold.
- **Technology:** Emails are sent via **SMTP**, with configurable Gmail account settings.
- **Interface:** Users can enable/disable alerts, set thresholds, specify recipient email, and test configuration directly from the dashboard.

2.5. User Interface (UI)

- **Framework:** Entirely built with **Streamlit**.
- **Navigation:** Sidebar contains dynamic filters for country, indicator, analysis period, and alert configuration.
- **Appearance:** Responsive layout with custom CSS, green theme, and simple animations.

3. Installation and Deployment

3.1. Prerequisites

- **Environment:** Python 3.9 or higher.
- **Python Dependencies:** Install via pip using requirements.txt:

pip install streamlit pandas numpy requests pymc arviz matplotlib plotly

- **SMTP Configuration:** A Gmail account with an app password is required for email alerts.

3.2. Launching the Application

1. Download project files, including dashboard.py.
2. Configure sender credentials in the send_email() function.
3. Run:

streamlit run dashboard.py

4. User Guide

4.1. Filter Configuration

- **Countries:** Select one or multiple Francophone African countries.
- **Indicators:** Choose "Renewable energy share (%)" or "Renewable electricity production (kWh)".
- **Years:** Define historical data range.
- **Forecast:** Set number of future years for Bayesian predictions.

4.2. Data Review

- Raw data displayed in interactive tables.
- Time series charts for trend comparison.
- CSV export button for downloading displayed data.

4.3. Bayesian Analysis Interpretation

- **Scope:** Analysis performed on the first selected country and first two indicators.
- **Results:** Forecast curves with 95% confidence intervals. Summary table of observed and predicted values.
- **Diagnostics:** Charts and statistics for evaluating predictive model quality.

4.4. Email Alert Management

- Enable alerts via the sidebar.
- Select country, indicator, and threshold.
- Enter recipient email and test configuration. Alerts are automatically sent if thresholds are exceeded.

5. Code Structure and Customization

File	Description
dashboard.py	Main Streamlit script containing UI logic, data collection, analysis, visualization, and alerts.
requirements.txt	Lists all Python dependencies.

Customizable parameters in send_email():

from_email = "your.email@gmail.com"

password = "your_app_password"

6. Future Enhancements

- **Data Sources:** Web scraping for additional data.

- **Analysis Scope:** Include more indicators and countries.
- **Advanced Alerts:** Multiple and scheduled alerts.
- **Exports:** PDF export functionality.
- **Deployment:** Host on cloud platforms (Heroku, AWS) for 24/7 access.

7. References and Resources

- PyMC Documentation
- Streamlit Documentation
- World Bank API Documentation