

Documentation Technique & Guide Utilisateur

1. Présentation du Projet

****Nom**** : Dashboard Veille Économique - Énergies Renouvelables Afrique Francophone

****Objectif**** : Fournir un outil interactif de collecte, visualisation et analyse bayésienne automatisée des données d'énergies renouvelables issues de la Banque Mondiale pour plusieurs pays d'Afrique francophone.

2. Architecture et Composants

2.1. Collecte des données

- API World Bank pour récupération automatique des indicateurs clés (ex. : part énergies renouvelables %, production électrique renouvelable).
- Utilisation de `fetch_worldbank_data()` avec cache Streamlit.

2.2. Analyse bayésienne

- Régression multivariée bayésienne avec PyMC.
- Prédictions futures paramétrables.
- Diagnostics : trace plots, R-hat, autocorrélations.

2.3. Visualisation

- Graphiques Plotly interactifs + Matplotlib.
- Résumé synthétique des données observées et prévisions.

2.4. Alertes

- Alerte email si un indicateur dépasse un seuil.
- SMTP Gmail configurable.
- Interface utilisateur intégrée.

2.5. Interface Utilisateur

- Streamlit + CSS personnalisé.
- Filtres dynamiques : pays, indicateurs, période, alertes.

3. Installation et Déploiement

3.1. Prérequis

```
```bash
pip install streamlit pandas numpy requests pymc arviz matplotlib plotly
````
```

3.2. Lancement

```
```bash
streamlit run dashboard.py
````
```

4. Guide Utilisateur

4.1. Filtres

- Choix des pays, indicateurs, années, alertes.

4.2. Visualisation

- Graphiques + tables interactives.

- Export CSV.

4.3. Analyse bayésienne

- Prévisions futures avec intervalle de crédibilité.
- Résumé + diagnostics.

4.4. Alertes email

- Activation dans la sidebar.
- Envoi email si seuil dépassé.

5. Fonctionnalités Avancées

- Scrapping à intégrer.
- Hébergement cloud possible.
- Export PDF ou envoi automatique régulier.

6. Structure du Code

| Fichier | Description |
|------------------|-----------------------------|
| ----- | ----- |
| dashboard.py | Script principal Streamlit. |
| requirements.txt | Dépendances Python. |

7. À personnaliser

Dans `send_email()` :

```
```python
from_email = "ton.email@gmail.com"
password = "ton_mdp_app"
````
```

8. Contact

****Auteur** :** [Ton Nom]

****Contact** :** [email@example.com]

Annexes

- PyMC : <https://docs.pymc.io>

- Streamlit : <https://docs.streamlit.io>

- API World Bank :

<https://datahelpdesk.worldbank.org/knowledgebase/articles/889386-api-documentation>