

# Guide d'utilisation

## Application Streamlit

Observatoire des prix à la consommation de la côte d'ivoire

TCHIMTCHOUA & GANAME

### ☰ Table des matières

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Fonctionnalités principales . . . . .	3
<b>2</b>	<b>Installation et configuration</b>	<b>3</b>
2.1	Prérequis système . . . . .	3
2.2	Téléchargement du projet . . . . .	4
<b>3</b>	<b>Méthodes de déploiement</b>	<b>4</b>
3.1	Méthode 1 : Environnement virtuel Python . . . . .	4
3.1.1	Création de l'environnement virtuel . . . . .	4
3.1.2	Installation des dépendances . . . . .	4
3.1.3	Lancement de l'application . . . . .	5
3.2	Méthode 2 : Déploiement Docker . . . . .	5
3.3	Accès à l'application . . . . .	5
<b>4</b>	<b>Architecture et navigation</b>	<b>6</b>
4.1	Structure de l'application . . . . .	6
4.2	Menu de navigation principal . . . . .	6
<b>5</b>	<b>Modules de visualisation</b>	<b>6</b>
5.1	Module Accueil . . . . .	6
5.2	Module Exploration . . . . .	7
5.3	Module Corrélations . . . . .	8
5.4	Module Choix (Modélisation) . . . . .	8
5.5	Module Prévisions . . . . .	9
5.6	Module Classification . . . . .	10
5.7	Module Alertes . . . . .	10
5.8	Module État Base . . . . .	10
<b>6</b>	<b>Modules d'outils</b>	<b>11</b>
6.1	Module Traitement . . . . .	11
6.1.1	Mode automatique . . . . .	11

6.1.2	Mode manuel . . . . .	11
6.2	Module Scraping ANSTAT . . . . .	11
<b>7</b>	<b>Guide d'utilisation avancée</b>	<b>12</b>
7.1	Workflow de modélisation . . . . .	12
7.2	Personnalisation de l'interface . . . . .	12
<b>8</b>	<b>Résolution des problèmes courants</b>	<b>12</b>
8.1	Problèmes de performance . . . . .	12
8.2	Problèmes de données . . . . .	12
<b>9</b>	<b>Architecture détaillée du projet</b>	<b>13</b>
9.1	Organisation modulaire . . . . .	13
9.1.1	Couche présentation . . . . .	13
9.1.2	Couche métier . . . . .	13
9.1.3	Couche données . . . . .	13
9.1.4	Outils d'analyse complémentaires . . . . .	13
9.2	Déploiement et conteneurisation . . . . .	14
<b>10</b>	<b>Conclusion et perspectives</b>	<b>15</b>
10.1	Capacités actuelles . . . . .	15
10.2	Évolutions futures . . . . .	15

## 1 Introduction

Cette application représente un **tableau de bord interactif** de dernière génération, développée avec le framework **Streamlit**, spécialement conçue pour le suivi, l'analyse et la surveillance de l'**Indice Harmonisé des Prix à la Consommation (IHPC)** en République de Côte d'Ivoire.

### Objectifs de l'application

L'Observatoire des prix offre une plateforme complète permettant aux analystes, économistes et décideurs de :

- Manipuler et traiter des données économiques complexes
- Visualiser les évolutions des prix de manière interactive
- Générer des prévisions économiques fiables
- Surveiller les anomalies et fluctuations du marché
- Exporter et partager les résultats d'analyses

### 1.1 Fonctionnalités principales

- **Analyse temporelle avancée** : Exploration des séries chronologiques avec tests statistiques
- **Modélisation prédictive** : Comparaison de modèles Prophet, SARIMA et Holt-Winters
- **Détection d'anomalies** : Système d'alertes automatisé pour les variations inhabituelles
- **Interface intuitive** : Navigation simplifiée avec mode sombre disponible
- **Export de données** : Génération de rapports et extraction de données

## 2 Installation et configuration

### 2.1 Prérequis système

#### Configuration recommandée

- **Python** : Version 3.11 ou supérieure
- **RAM** : Minimum 4 GB (8 GB recommandés)
- **Espace disque** : 2 GB d'espace libre
- **Connexion internet** : Requête pour le scraping ANSTAT

## 🔧 2.2 Téléchargement du projet

Pour obtenir le code source de l'application, utilisez Git pour cloner le dépôt officiel :

```
git clone https://github.com/GanItachi/ObservatoirePrix.git cd
ObservatoirePrix
```

### 📄 Structure du projet

Le dépôt cloné contient :

- `app.py` : Point d'entrée principal de l'application
- `app_pages/` : Modules des différentes pages
- `data/` : Fichiers de données et sauvegardes
- `requirements.txt` : Liste des dépendances Python
- `Dockerfile` : Configuration pour le déploiement containerisé

## 📈 3 Méthodes de déploiement

### 🔧 3.1 Méthode 1 : Environnement virtuel Python

Cette méthode est recommandée pour un développement local et un contrôle précis des dépendances.

#### ➔ 3.1.1 Création de l'environnement virtuel

```
# Création de l'environnement virtuel
python -m venv Observ
# Activation (Windows)
Observ \ Scripts \ activate
```

#### ➔ 3.1.2 Installation des dépendances

```
Installation des packages requis
pip install -r requirements.txt
```

### 📄 Temps d'installation

L'installation complète des dépendances peut prendre entre 5 et 15 minutes selon votre connexion internet et la puissance de votre machine.

### → 3.1.3 Lancement de l'application

```
Démarrage du serveur Streamlit  
streamlit run app.py
```

## ⚙️ 3.2 Méthode 2 : Déploiement Docker

Docker offre une solution plus rapide et plus portable, garantissant un environnement d'exécution identique.

### ✓ Avantages de Docker

- Installation automatisée de toutes les dépendances
- Environnement isolé et reproductible
- Déploiement simplifié sur différentes plateformes
- Rechargement automatique du code

```
Construction et lancement du conteneur  
docker compose up -build
```

## ⚙️ 3.3 Accès à l'application

Une fois l'application lancée, elle sera accessible via votre navigateur à l'adresse :

 <http://localhost:8501/>

## 4 Architecture et navigation

### 4.1 Structure de l'application

L'application suit une architecture modulaire organisée autour du fichier principal `app.py` qui :

1. Charge automatiquement la base de données `Base.csv`
2. Initialise les variables de configuration (postes, fonctions, agrégats)
3. Configure l'interface utilisateur avec la barre latérale
4. Gère la navigation entre les différents modules
5. Active les fonctionnalités de thème (mode sombre/clair)

### 4.2 Menu de navigation principal

L'interface est organisée en deux sections principales :

Section	Modules disponibles
Visualisation	Accueil, Exploration, Corrélations, État Base, Choix, Prévisions, Classification, Alertes
Outils	Traitement, Scraping ANSTAT

TABLE 1 – Organisation du menu de navigation

## 5 Modules de visualisation

### 5.1 Module Accueil

#### Vue d'ensemble économique

Le module d'accueil offre un **tableau de bord synthétique** présentant :

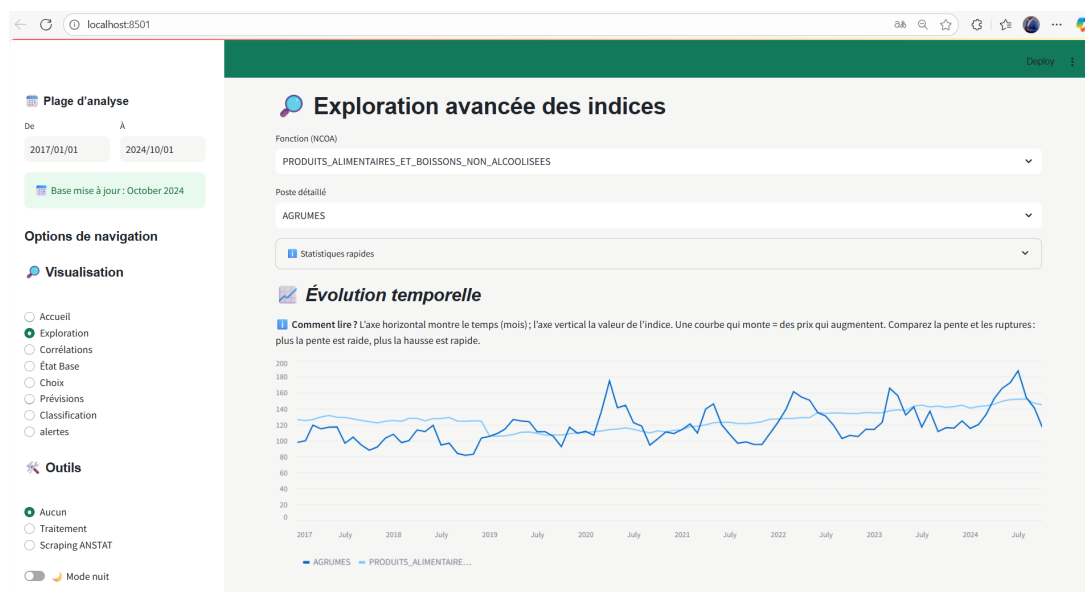
- Indicateurs clés : IHPC actuel, taux d'inflation
- Variations périodiques (mensuelle, trimestrielle, annuelle)
- Graphiques de tendance interactive
- Fonctionnalité d'export des données au format CSV



## 5.2 Module Exploration

Ce module permet une **analyse approfondie** d'un poste spécifique. Pour se faire il est possible de choisir une fonction de la NCOA, et de choisir le poste à analyser en son sein.

- **Visualisations temporelles** : Courbes d'évolution, tendances saisonnières
- **Tests statistiques** :
  - Test de saisonnalité
  - Test de Dickey-Fuller pour la stationnarité
  - Analyse de la distribution
- **Représentations graphiques** : Cartes de chaleur, diagrammes en boîte, histogrammes

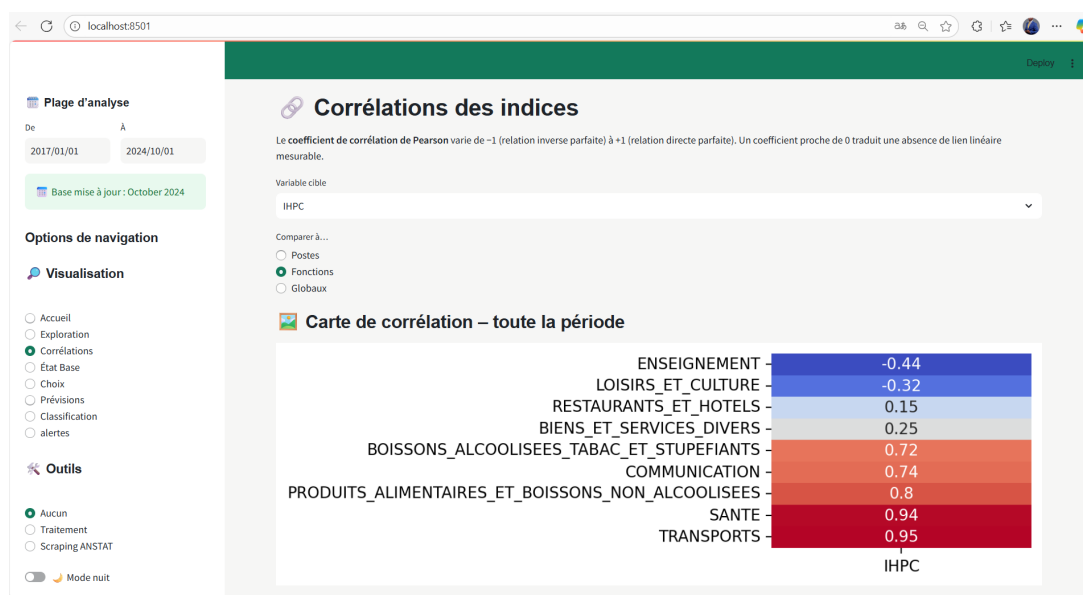


## 5.3 Module Corrélations

### ✓ Analyse des relations

Calcul et visualisation des **corrélations de Pearson** entre variables :

- Matrice de corrélation interactive (heatmap)
- Classement des corrélations les plus fortes/faibles
- Identification des variables liées économiquement



## 5.4 Module Choix (Modélisation)

### i Comparaison de modèles

Évaluation comparative des algorithmes de prévision :

- **Prophet** : Modèle développé par Facebook, excellent pour les tendances
- **SARIMA** : Modèle classique intégrant saisonnalité et auto-régression
- **Holt-Winters** : Méthode de lissage exponentiel avec composantes saisonnières

Critères d'évaluation : RMSE, MAE, MAPE avec possibilité de sauvegarde du meilleur modèle.

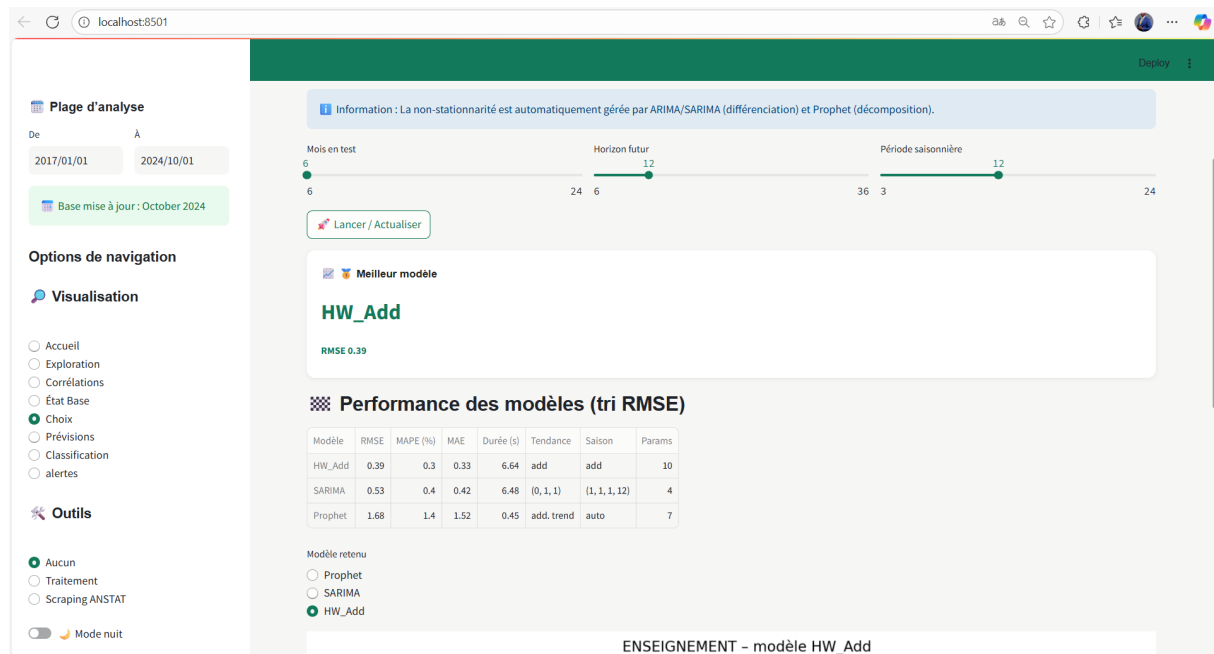
NB : Un modèle doit être enregistré pour pouvoir l'utiliser ultérieurement pour des prévision, chaque enregistrement efface l'ancien. Le bouton '**Enregistré**' se trouve tout en bas



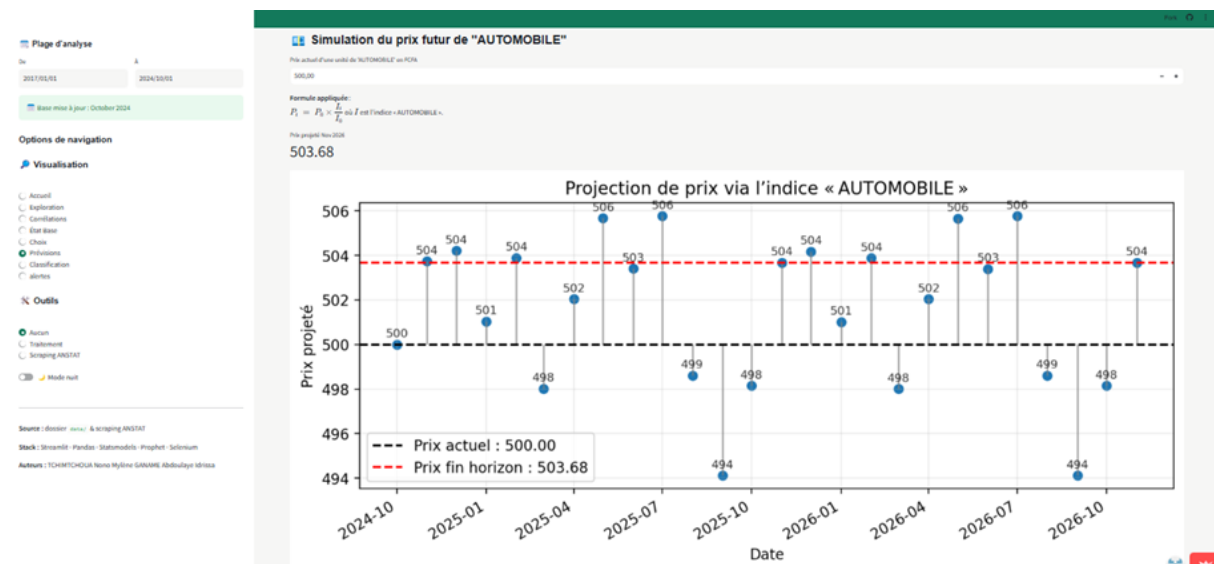
## 5.5 Module Prévisions

Utilisation du **modèle de prévision enregistré** pour :

- Projection des indices à court et moyen terme La date de projection choisie, une prediction est faite sur l'évolution de l'indice choisi dans le choix du modèle.



- Simulation de l'impact sur les prix réels L'application ne travail que sur des indices. Mais la possibilité est offerte de voir l'évolution des prix en renseignant le prix actuel du poste concerné.



## 5.6 Module Classification

**Typologie automatisée** des postes économiques basée sur :

- Clustering pré-calculé des comportements de prix (tendances, saisonnalité, volatilité ...)
- Visualisation des groupes homogènes
- Export des classifications pour analyses externes

## 5.7 Module Alertes

### ⚠ Système de surveillance

Détection automatique des **anomalies et ruptures** dans les séries :

- Configuration des seuils d'alerte personnalisés
- Paramétrage de la fenêtre d'observation
- Analyse de la volatilité des prix
- Export des alertes pour suivi externe

Les alertes varient selon les seuils que vous aurez définis pour chaque variable.

## 5.8 Module État Base

**Diagnostic complet** de la base de données chargée :

- Statistiques descriptives (nombre de lignes, colonnes)
- Période couverte par les données
- Analyse des valeurs manquantes
- Qualité et complétude des informations

## 6 Modules d'outils

### 6.1 Module Traitement

Ce module offre des fonctionnalités avancées de **nettoyage et de préparation** des données :

#### → 6.1.1 Mode automatique

- Remplacement des valeurs NaN par 100 (base de référence)
- Chaînage automatique des indices depuis 2019
- Application des règles de cohérence standard

#### → 6.1.2 Mode manuel

- Choix de la méthode d'imputation (moyenne, médiane, interpolation)
- Sélection de la date pivot pour le chaînage
- Intégration de compléments NCOA ( Toutes les fonctions ne sont pas automatiquement disponible sur les sources utilisées)
- Validation manuelle des corrections

#### Sauvegarde automatique

Toutes les modifications sont automatiquement sauvegardées dans le dossier **data/backups/** avec horodatage pour traçabilité.

### 6.2 Module Scraping ANSTAT

**Automatisation de la collecte** de données depuis le site officiel de l'ANSTAT :

- Téléchargement des fichiers CSV récents en cliquant sur '*Lancer le scrapping*'
- La fusion de toutes les nouvelles données collectées en une seule base de données chargé automatiquement au projet démarrage.

#### Connexion requise

Cette fonctionnalité nécessite une connexion internet stable.

## 7 Guide d'utilisation avancée

---


### Configuration de l'analyse temporelle

**Sélection de la période** : Utilisez le sélecteur temporel dans la barre latérale.

#### 7.1 Workflow de modélisation

1. **Préparation** : Nettoyez les données via le module Traitement
2. **Exploration** : Analysez les caractéristiques des séries
3. **Modélisation** : Comparez les modèles dans le module Choix
4. **Validation** : Évaluez les performances sur données de test
5. **Sauvegarde** : Enregistrez le meilleur modèle pour utilisation future
6. **Prévision** : Générez les projections via le module Prévisions

#### 7.2 Personnalisation de l'interface

- **Mode sombre** : Activez via l'interrupteur  pour un confort visuel optimal
- **Largeur d'affichage** : Ajustez automatiquement selon la taille de votre écran
- **Export de données** : Formats disponibles : CSV, Excel, PDF (graphiques)

## 8 Résolution des problèmes courants

---

#### 8.1 Problèmes de performance

- **Lenteur de l'application** : Réduisez la période d'analyse ou le nombre de variables
- **Erreurs de mémoire** : Redémarrez l'application, vérifiez la RAM disponible
- **Graphiques non affichés** : Actualisez la page, vérifiez votre navigateur

#### 8.2 Problèmes de données

- **Données manquantes** : Utilisez le module Traitement pour l'imputation
- **Incohérences** : Vérifiez l'état de la base via le module État Base

## 9 Architecture détaillée du projet

---

### 9.1 Organisation modulaire

L'Observatoire des prix suit une architecture logicielle moderne basée sur la séparation des responsabilités :

#### → 9.1.1 Couche présentation

- **app.py** : Point d'entrée principal orchestrant l'interface Streamlit
- **design.py** : Configuration et thèmes de l'interface utilisateur
- **app\_pages/** : Modules spécialisés pour chaque fonctionnalité
  - **accueil.py** : Tableau de bord principal
  - **exploration.py** : Analyse des séries temporelles
  - **alertes.py** : Système de surveillance des anomalies
  - **correlations.py** : Analyse des corrélations
  - **previsions.py** : Module de prévision

#### → 9.1.2 Couche métier

- **core/** : Cœur algorithmique de l'application
  - **models.py** : Implémentation des modèles ARIMA/SARIMA, Prophet, clustering
  - **preprocessing.py** : Traitement des valeurs manquantes et chaînage des indices
  - **nomenclature.py** : Gestion des postes selon la nomenclature NCOA

#### → 9.1.3 Couche données

- **data/** : Stockage des données traitées
  - **Base.csv** : Base principale des indices agrégés
  - **Base\_long.csv** : Format tidy pour l'exploration statistique
  - **backups/** : Sauvegardes versionnées automatiques et manuelles
- **excels\_anstat\_new/** : Données brutes extraites par scraping
- **telechargements\_temp/** : Zone de transit pour le téléchargement

#### → 9.1.4 Outils d'analyse complémentaires

- **Clustering.ipynb** : Notebook Jupyter pour l'analyse typologique avancée
- **desc\_clusters.csv** : Description détaillée des profils de clusters
- **mapping\_clusters.csv** : Correspondance visuelle des groupes

## ⚙️ 9.2 Déploiement et conteneurisation

### ✓ Docker et orchestration

L'application inclut une configuration complète pour le déploiement :

- **Dockerfile** : Image containerisée de l'application
- **docker-compose.yml** : Orchestration des services
- **requirements.txt** : Gestion des dépendances Python

Cette approche garantit la reproductibilité et facilite le déploiement sur différents environnements.

## 10 Conclusion et perspectives

### 10.1 Capacités actuelles

L'Observatoire des prix offre aujourd'hui une solution complète pour :

- **Analyse économique** : Suivi en temps réel des indicateurs de prix
- **Prévision** : Modélisation avancée pour l'aide à la décision
- **Surveillance** : Détection précoce des anomalies de marché
- **Reporting** : Génération automatisée de rapports d'analyse

### 10.2 Évolutions futures

#### Roadmap de développement

- Intégration d'algorithmes d'intelligence artificielle avancés
- Connexion API avec les bases de données officielles
- Module de géolocalisation des prix par région
- Interface mobile responsive
- Système de notifications en temps réel



**Application accessible à :**

**[https://observatoireprix-civ.  
streamlit.app/](https://observatoireprix-civ.streamlit.app/)**



**Code source :**

**<https://github.com/GanItachi/ObservatoirePrix>**