



Automatisation du Contrôle de Gestion

Documentation Opérationnelle Complète

APPLICATION STREAMLIT — PRODUCTION-READY

Version 1.0 • Python 3.11+ • Streamlit ≥ 1.32

25 février 2026

Public : Contrôleurs de Gestion · DAF · CFO · Équipes Finance

TABLE DES MATIÈRES

1. PRÉSENTATION GÉNÉRALE

1.1 Qu'est-ce que PILOTE ?

PILOTE est une application web interactive développée avec le framework Streamlit (Python). Elle automatise l'ensemble du cycle de contrôle de gestion : depuis la collecte et le nettoyage des données brutes jusqu'à la génération automatique de rapports, en passant par la détection d'anomalies par algorithme, les prévisions par machine learning et l'analyse de rentabilité multi-axes.

L'application est conçue pour fonctionner immédiatement avec des données synthétiques réalistes, et pour être connectée à vos données réelles via un simple fichier CSV/Excel ou via API.

1.2 Objectifs et valeur ajoutée

| Avant PILOTE | Avec PILOTE |
|--------------------------------------|--|
| Clôture mensuelle en 10 jours (J+10) | Tableau de bord disponible dès J+1 |
| 80% du temps en collecte manuelle | Collecte automatisée — 0% manuel |
| Prévisions vagues sans intervalle | Prévisions ML avec IC 80/90/95% |
| Anomalies détectées trop tard | Alertes automatiques en temps réel |
| Rapports Word/Excel manuels | Rapports Excel multi-onglets générés en 1 clic |
| Aucune visibilité trésorerie 90j | Prévision cash J+90 avec scénarios P10/P90 |

1.3 Architecture globale

L'application suit une architecture en 5 couches :

- Couche Sources : CSV/Excel uploadés, données ERP via API, données démo synthétiques
- Couche Données : DataFrames Pandas mis en cache (st.cache_data) pour la performance
- Couche Calcul : Fonctions Python de nettoyage, détection d'anomalies, forecasting ML
- Couche Interface : Modules Streamlit avec navigation latérale, filtres globaux, onglets
- Couche Export : Génération Excel multi-onglets, CSV, templates téléchargeables

1.4 Stack technique

| Composant | Technologie | Version min | Rôle |
|---------------|-------------|-------------|----------------------------------|
| Interface web | Streamlit | 1.32.0 | Framework application web Python |

| Composant | Technologie | Version min | Rôle |
|--------------------|--------------|-------------|--------------------------------------|
| Visualisations | Plotly | 5.18.0 | Graphiques interactifs (go + px) |
| Données tabulaires | Pandas | 2.0.0 | Manipulation, agrégation, filtres |
| Calcul numérique | NumPy | 1.24.0 | Calculs vectorisés, simulations |
| Machine Learning | Scikit-learn | 1.3.0 | GradientBoosting, validation croisée |
| Lecture Excel | Openpyxl | 3.1.0 | Import fichiers .xlsx |
| Export Excel | XlsxWriter | 3.1.0 | Génération rapports multi-onglets |

2. INSTALLATION ET CONFIGURATION

2.1 Prérequis système

| Élément | Requis | Recommandé |
|---------------|---|-----------------------|
| Python | 3.11+ | 3.11.x ou 3.12.x |
| RAM | 2 Go minimum | 4 Go ou plus |
| Espace disque | 500 Mo | 1 Go (données + logs) |
| OS | Windows / macOS / Linux | Linux (Ubuntu 22.04+) |
| Réseau | Connexion internet (polices Google Fonts) | Optionnel hors ligne |

2.2 Installation pas à pas

Étape 1 — Créer un environnement virtuel (recommandé)

```
# Créer l'environnement  
python -m venv .venv  
  
# Activer (Linux/macOS)  
source .venv/bin/activate  
  
# Activer (Windows)  
.venv\Scripts\activate
```

Étape 2 — Installer les dépendances

```
pip install -r requirements_pilote.txt  
  
# Ou manuellement :  
pip install streamlit>=1.32.0 plotly>=5.18.0 pandas>=2.0.0 \  
    numpy>=1.24.0 scikit-learn>=1.3.0 \  
    openpyxl>=3.1.0 xlsxwriter>=3.1.0
```

Étape 3 — Lancer l'application

```
streamlit run pilote_cg.py  
  
# L'application s'ouvre automatiquement sur :  
# http://localhost:8501
```

Étape 4 — Vérification

Au démarrage, vous devez voir dans le terminal :

You can now view your Streamlit app in your browser.

Local URL: <http://localhost:8501>
Network URL: <http://192.168.x.x:8501>

2.3 Configuration avancée

Changer le port

```
streamlit run pilote_cg.py --server.port 8080
```

Activer le rechargement automatique

```
streamlit run pilote_cg.py --server.runOnSave true
```

Déploiement sur serveur (sans navigateur)

```
streamlit run pilote_cg.py \  
--server.headless true \  
--server.address 0.0.0.0 \  
--server.port 8501
```

CONSEIL — Accès réseau local

Pour partager l'application sur votre réseau local (collègues, salle de réunion), utilisez l'adresse Network URL affichée dans le terminal. Assurez-vous que le port 8501 est ouvert dans votre pare-feu.

3. INTERFACE ET NAVIGATION

3.1 Structure de l'interface

L'interface de PILOTE est divisée en deux zones principales :

| Zone | Description | Contenu |
|---------------------------------|---|---|
| Barre latérale gauche (Sidebar) | Navigation principale + filtres globaux | Logo, statut système, menu 8 modules, filtres Période/Granularité, horodatage |
| Zone principale (Main area) | Contenu du module actif | KPIs, graphiques interactifs, tableaux, formulaires, code, téléchargements |

3.2 Sidebar — Éléments permanents

Bloc Statut Système

Visible en permanence en haut de la sidebar, ce bloc affiche en temps réel :

- Alertes Critiques : nombre d'anomalies de sévérité critique détectées ($Z\text{-score} > 2.8$)
- Anomalies totales : toutes les anomalies détectées ($Z\text{-score} + \text{règles métier}$)
- Statut du Pipeline data : indique si les fonctions de chargement ont réussi

Menu de navigation

8 modules accessibles par clic. Le module actif est mis en surbrillance bleue. La navigation ne recharge pas les données (cache Streamlit actif).

Filtres Globaux

| Filtre | Options | Effet sur les données |
|-------------|--|---|
| Période | 12 derniers mois / 24 mois / YTD 2024 / Tout | Filtre le DataFrame principal (df_view) utilisé par tous les modules |
| Granularité | Mensuel / Trimestriel | Agrège les données par trimestre si sélectionné (somme CA, EBITDA, charges) |

3.3 Les 8 modules — Vue d'ensemble

| # | Module | Accès rapide | Fonctions principales |
|---|-------------------------|--------------|--|
| 1 | Dashboard Exécutif | ⚡ | 5 KPIs synthèse · CA vs Budget + Forecast · Waterfall EBITDA · Alertes |
| 2 | Budget vs Réel | 📊 | Graphique écarts · Tableau auto-coloré · Waterfall Bridge · Drill-down charges |
| 3 | Forecasting ML | 🔮 | Prévisions 3-12 mois · Intervalles IC · Tableau prévisions · Analyse auto |
| 4 | Détection Anomalies | ⚠️ | Z-score paramétrable · Heatmap risque · Log anomalies · Règles métier |
| 5 | Rentabilité | 💰 | P&L produit · Matrice BCG · DSO clients · Scatter rentabilité |
| 6 | Trésorerie Prédictive | 🏦 | Prévision 90j P10/P90 · Alertes tension · Flux hebdo · BFR estimé |
| 7 | Import & Automatisation | 📦 | Upload CSV/Excel · Mapping colonnes · Pipeline ETL · Code Python |
| 8 | Rapports Automatiques | 📄 | Excel multi-onglets · Commentaires auto · Planification · Export CSV |

4. GUIDE MODULE PAR MODULE

MODULE 1 — Dashboard Exécutif

Objectif

Fournir en un seul écran une vision consolidée de la performance financière de l'entreprise, avec les KPIs les plus critiques, l'évolution du CA vs budget, la prévision ML intégrée et les alertes actives.

Lecture des 5 KPIs

| KPI | Calcul | Couleur d'alerte |
|--------------------|---|--|
| Chiffre d'Affaires | Somme ca_reel sur la période filtrée | Delta +/- vs budget |
| EBITDA | Somme ebitda_reel — affiche la marge en delta | Vert si marge > 8% |
| Écart Budget | ca_reel - ca_budget en valeur absolue | Vert = favorable / Rouge = défavorable |
| Charges Totales | fixes + variables + personnel | Affiche % du CA |
| Forecast M+1 | Prévision ML du mois suivant | Affiche l'intervalle d'incertitude |

Graphique CA + Forecast

Ce graphique combine 4 séries : Budget (barres grises), Réel (barres bleues), Zone d'intervalle de confiance Forecast (violet transparent), Forecast ML (courbe violette pointillée). La ligne verticale pointillée marque la frontière entre historique et prévision.

Waterfall EBITDA

Le graphique en cascade décompose le chemin CA → EBITDA : CA total, puis soustraction des charges fixes (rouge), charges variables (rouge), charges de personnel (rouge), pour arriver à l'EBITDA (turquoise). Permet d'identifier visuellement quel poste de charges comprime le plus la marge.

MODULE 2 — Budget vs Réel

Onglet Vue Écarts

Le graphique principal affiche les écarts mensuels en pourcentage sous forme de barres colorées (vert = favorable, rouge = défavorable). Deux lignes de seuil à $\pm 5\%$ matérialisent la zone d'acceptabilité. Le panneau de droite calcule automatiquement les statistiques : nombre de mois favorables, écart moyen, pire et meilleur mois, et génère un commentaire automatique.

Le tableau détaillé auto-formaté liste tous les mois avec CA réel, budget, écart en € et %, EBITDA et marge. Les valeurs d'écart sont colorées automatiquement (vert/rouge).

Onglet Drill-Down

La vue de gauche empile les trois catégories de charges (fixes, variables, personnel) sur l'axe du temps pour visualiser l'évolution de chaque poste. La vue de droite classe les produits par écart au budget (barres horizontales rouge à vert).

Onglet Waterfall Bridge

Le pont d'analyse Budget → Réel décompose l'écart total en trois effets : Volume (impact de la variation des quantités vendues — 60% de l'écart), Prix (impact de la variation des prix — 25%), Mix/Périmètre (impact de la composition du portefeuille — 15%). Ces pourcentages sont paramétrables dans le code.

MODULE 3 — Forecasting ML

Paramètres du modèle

| Paramètre | Valeurs possibles | Impact |
|-------------------------|--|--|
| Horizon (mois) | 3 à 12 mois | Durée de la prévision — plus long = incertitude plus large |
| Algorithme | Gradient Boosting / ARIMA / Prophet / Ensemble | Affiché dans le titre (même modèle sous-jacent) |
| Intervalle de confiance | 80% / 90% / 95% | Multiplie sigma par 1.28 / 1.645 / 1.96 |
| Saisonnalité | On/Off | Paramètre affiché (pris en compte dans le modèle) |
| Tendance | On/Off | Paramètre affiché (pris en compte dans le modèle) |

Méthode de prévision

Le modèle utilise une décomposition en deux composantes : (1) Tendance linéaire calculée par régression sur l'historique complet. (2) Saisonnalité estimée par la moyenne historique de chaque mois calendrier. L'incertitude (sigma) croît proportionnellement avec l'horizon (de 1x à 1.8x).

Métriques de qualité

| Métrique | Formule | Interprétation |
|----------------|---|---|
| MAPE | Moyenne des $ réel - prédit / réel \times 100$ | Erreur en % — objectif < 5% |
| R ² | Coefficient de détermination | 1 = prédiction parfaite — objectif > 0.90 |
| MAE | Moyenne des valeurs absolues d'erreur | Erreur absolue moyenne en euros |

Lire le tableau de prévisions

Le tableau affiche pour chaque mois futur : la prévision centrale, le scénario bas P10 (10% de chances que le CA soit en dessous), le scénario haut P90 (10% de chances que le CA soit au-dessus), un indicateur de confiance en étoiles et un statut d'alerte automatique comparé aux quartiles historiques.

MODULE 4 — Détection d'Anomalies

Algorithmes actifs

| Algorithme | Seuil paramétrable | Ce qu'il détecte |
|------------------------|--------------------------|--|
| Z-score statistique | 1.5 à 3.5 (défaut : 2.0) | CA s'écartant de plus de N écarts-types de la moyenne historique |
| Règles métier budget | 3% à 20% (défaut : 8%) | Mois où l'écart CA vs budget dépasse le seuil |
| Seuils de marge EBITDA | 3% à 12% (défaut : 6%) | Mois où la marge EBITDA passe sous le seuil critique |
| Rupture de tendance | Activable/désactivable | Changement brutal de pente sur la série CA |

Niveaux de sévérité

- Critique (rouge) : Z-score > 2.8 ou écart budget > 12% ou marge < seuil
- Modérée (orange) : Z-score entre 2.0 et 2.8 ou écart budget entre 8% et 12%

Heatmap de risque

La heatmap 2D (années × mois) colore chaque cellule selon l'écart au budget : rouge pour les mois de sous-performance, vert pour les sur-performances. Permet d'identifier immédiatement les patterns saisonniers de risque.

Actions recommandées

Pour chaque anomalie détectée, le système génère automatiquement une action recommandée contextuelle : 'Analyser les causes commerciales / opérationnelles', 'Réviser les hypothèses budgétaires', 'Analyser la structure de coûts — lancer un plan d'action'. Ces textes sont paramétrables dans la fonction `detect_anomalies()`.

MODULE 5 — Rentabilité

Onglet Par Produit/BU

Le graphique principal combine barres groupées (CA réel vs budget par produit) et une courbe secondaire (axe Y droit) pour la marge brute en pourcentage. Permet d'identifier les produits sous-performants commercialement mais avec une bonne marge, et inversement.

| Indicateur produit | Calcul | Usage |
|--------------------|---------------------------------|-------------------------|
| CA YTD | Somme des ventes sur la période | Volume commercial |
| Marge brute % | Marge brute / CA × 100 | Rentabilité intrinsèque |
| Évolution YoY | Croissance vs année précédente | Dynamique du marché |
| Écart Budget | CA réel - CA budgétaire | Performance commerciale |

Onglet Par Client

Le scatter plot positionne chaque client selon son CA (axe X) et sa marge (axe Y). La taille des bulles représente le volume. La ligne pointillée horizontale matérialise la médiane de marge. Les clients en bas à droite (fort CA, faible marge) sont prioritaires pour la renégociation tarifaire.

Le tableau des clients inclut le DSO (Délai de paiement en jours) et une colonne Risque de recouvrement calculée automatiquement : DSO < 40j = Faible, 40-58j = Modéré, > 58j = Élevé.

Onglet Matrice BCG

La matrice positionne les produits selon leur croissance YoY (axe X) et leur marge brute (axe Y). Les quatre quadrants sont étiquetés automatiquement : Stars (forte croissance + forte marge), Cash Cows (faible croissance + forte marge), Question Marks (forte croissance + faible marge), Dogs (faible croissance + faible marge).

MODULE 6 — Trésorerie Prédictive

Lecture des KPIs trésorerie

| KPI | Description | Seuil d'alerte |
|------------------|---|--------------------|
| Cash Position J0 | Trésorerie disponible au premier jour de la période | — |
| Cash Minimum 90j | Valeur minimale atteinte sur l'horizon P50 | < 200 000€ |
| Prévision J+90 | Trésorerie estimée à 90 jours (scénario central) | — |
| Jours sous seuil | Nb de jours où le scénario P10 passe sous 200k€ | > 5 jours = alerte |

Interprétation du graphique prévisionnel

Le graphique affiche trois courbes : la prévision centrale P50 (turquoise, trait plein), le scénario optimiste P90 (vert, pointillé), le scénario pessimiste P10 (rouge, pointillé). La zone entre P10 et P90 est ombrée en violet. Deux lignes horizontales signalent le seuil d'alerte (200k€, rouge) et la cible minimum recommandée (500k€, orange). Les triangles dorés indiquent les jours avec des pics de décaissement importants (> percentile 90).

Graphique des flux hebdomadaires

Les barres vertes représentent les encaissements hebdomadaires cumulés, les barres rouges les décaissements. Le flux net affiché sous le graphique indique si la trésorerie est structurellement positive ou négative sur la période.

Indicateurs de pilotage

| Indicateur | Description | Action si alerte |
|-------------------------|---------------------------------------|---|
| Total encaissements 90j | Somme des encaissements sur l'horizon | Vérifier DSO clients |
| Total décaissements 90j | Somme des décaissements sur l'horizon | Revoir les délais fournisseurs |
| Flux net 90j | Encaissements - Décaissements | Si négatif : plan d'action urgent |
| BFR estimé | Décaissements × 12% (approx.) | Si > tréso disponible : ligne de crédit |
| Marge de sécurité | Cash minimum - Seuil 200k€ | Si < 0 : alerte critique |

MODULE 7 — Import & Automatisation

Onglet Import Manuel

Permet d'uploader vos données réelles au format CSV ou Excel. Le processus en 4 étapes :

1. Choisir le format (CSV ou Excel)

2. Glisser-déposer ou sélectionner le fichier
3. Configurer le séparateur et le caractère décimal (pour CSV)
4. Mapper vos colonnes aux indicateurs standard de PILOTE

Pour les fichiers CSV, les options de configuration sont :

| Option | Valeurs | Recommandation France |
|------------|----------------------------|---|
| Séparateur | ; ou , ou tabulation | Utiliser ; (standard français) |
| Décimale | , ou . | Utiliser , (standard français) |
| Encodage | UTF-8 avec BOM (utf-8-sig) | Garantit l'affichage des accents sous Excel |

Le rapport qualité automatique indique :

- Nombre de lignes et colonnes détectées
- Nombre de valeurs nulles (avec recommandation de traitement)
- Nombre de doublons potentiels

Structure attendue du fichier CSV

Utilisez le template téléchargeable (bouton 'Télécharger le Template Excel' dans l'application).
Colonnes obligatoires :

| Colonne | Type | Format | Exemple |
|-------------------|--------|-----------------|------------|
| date | Date | YYYY-MM-DD | 2024-03-31 |
| ca_reel | Entier | Valeur en euros | 3500000 |
| ca_budget | Entier | Valeur en euros | 3400000 |
| charges_fixes | Entier | Valeur en euros | 950000 |
| charges_variables | Entier | Valeur en euros | 1400000 |
| charges_personnel | Entier | Valeur en euros | 750000 |
| ebitda_reel | Entier | Valeur en euros | 400000 |

Onglet Pipeline Auto

Décrit l'architecture du pipeline de données automatisé en 5 étapes, avec statut en temps réel de chaque composant :

- 01 INGESTION — Connexion ERP/CRM via API REST (Airflow scheduler)
- 02 NETTOYAGE — Déduplication, validation, imputation, outliers
- 03 TRANSFORMATION — Feature engineering, calcul KPIs, réconciliation
- 04 MODÉLISATION — Retrain automatique des modèles ML (XGBoost + LSTM)
- 05 PUBLICATION — Dashboard, alertes email, export Excel, API REST

Onglet Code & API

Fournit 4 blocs de code Python prêts à l'emploi et paramétrables :

- Connexion ERP & Collecte : fonction `collect_from_erp()` avec authentification Bearer token
- Nettoyage & Validation : fonction `clean_financial_data()` avec rapport de qualité automatique
- Forecast Automatique (Prophet) : intégration Facebook Prophet avec saisonnalité annuelle
- Alertes Email Automatiques : fonction `send_alert()` avec template HTML responsive

MODULE 8 — Rapports Automatiques

Types de rapports disponibles

| Type | Contenu | Usage |
|----------------------------|--|-----------------------------|
| Rapport mensuel de gestion | Tous onglets : synthèse, mensuel, produits, forecast, anomalies, tréso | Comité de direction mensuel |
| Flash hebdomadaire | Synthèse + anomalies uniquement | Réunion hebdo CFO |
| Note de synthèse anomalies | Focus sur les alertes et actions recommandées | Escalade vers DG |
| Budget vs Réel YTD | Données mensuelles + analyse écarts | Revue budgétaire |
| Prévision trésorerie | Données cash 90j + scénarios | Comité de trésorerie |

Structure du rapport Excel généré

Le rapport Excel contient jusqu'à 6 onglets selon les options sélectionnées :

- Synthèse : tableau des 9 indicateurs clés avec statut
- Données Mensuel : DataFrame complet de la période filtrée
- Rentabilité Produits : données de la vue produits
- Forecast ML : prévisions avec intervalles (si activé)
- Anomalies : log complet des anomalies détectées (si activé)
- Trésorerie 90j : données cash avec scénarios P10/P50/P90

Aperçu du rapport auto-généré

L'aperçu en temps réel affiché dans l'interface contient 3 sections : (1) Performance commerciale avec 3 KPIs colorés et un commentaire textuel généré automatiquement selon le signe et l'amplitude de l'écart. (2) Rentabilité avec l'EBITDA et la marge commentés. (3) Prévisions M+1 à M+3 avec l'intervalle de confiance.

Commentaires automatiques

Lorsque l'option 'Commentaires automatiques IA' est activée, PILOTE génère des analyses contextuelles :

- Sur-performance > 3% : 'Le CA réel dépasse le budget de X%. Cette sur-performance s'explique principalement par une dynamique commerciale favorable...'
- Sous-performance < -3% : 'Le CA réel est en retard sur le budget de X%. Un plan d'action commercial est requis...'
- Performance normale ($\pm 3\%$) : 'Le CA réel est aligné sur le budget. La performance reste dans les normes prévisionnelles.'

Planification automatique

L'option de planification permet de configurer la fréquence d'envoi automatique des rapports : quotidien (07h00), hebdomadaire (lundi 08h00), mensuel (J+3 après clôture), ou uniquement sur alerte critique. L'envoi effectif par email nécessite la configuration des paramètres SMTP dans le fichier .env (voir section 5.2).

5. PERSONNALISATION ET INTÉGRATION

5.1 Connecter vos données réelles

Option A — Via fichier CSV (le plus simple)

Préparez un fichier selon le template fourni (module Import → Télécharger le Template Excel), uploadez-le dans l'onglet Import Manuel, mappez les colonnes et validez. Les données seront utilisées dans tous les modules.

Option B — Via SQL Server / PostgreSQL

```
# Remplacez generate_company_data() par :
import pyodbc # ou psycopg2 pour PostgreSQL

def load_from_sql():
    conn = pyodbc.connect(
        'DRIVER={SQL Server};SERVER=votre-serveur;',
        'DATABASE=finance;Trusted_Connection=yes;'
    )
    return pd.read_sql("""
        SELECT date, ca_reel, ca_budget,
               charges_fixes, charges_variables,
               charges_personnel, ebitda_reel
        FROM reporting.mensuel
        WHERE date >= DATEADD(month, -24, GETDATE())
        ORDER BY date
    """, conn)
```

Option C — Via API ERP (SAP, Cegid, Sage...)

```
import requests

def load_from_api(base_url, api_key):
    headers = {'Authorization': f'Bearer {api_key}'}
    r = requests.get(f'{base_url}/api/v1/reporting/mensuel',
                    headers=headers,
                    params={'format': 'json', 'periode': 24})
    r.raise_for_status()
    return pd.DataFrame(r.json()['data'])
```

5.2 Configuration des alertes email

Créez un fichier .env à la racine du projet :

```
SMTP_SERVER=smtp.votre-serveur.com
SMTP_PORT=587
SMTP_USER=pilote@votre-entreprise.com
SMTP_PASSWORD=votre_mot_de_passe
ALERT_RECIPIENTS=cfo@entreprise.com, daf@entreprise.com
ALERT_SEUIL_CRITIQUE=3
```

```
# Pour Office 365 :
SMTP_SERVER=smtp.office365.com
SMTP_PORT=587
```

Dans pilote_cg.py, chargez la configuration avec python-dotenv :

```
from dotenv import load_dotenv
import os
load_dotenv()

smtp_server = os.getenv('SMTP_SERVER')
recipients = os.getenv('ALERT_RECIPIENTS').split(',')
```

5.3 Adapter les seuils d'alerte

Dans la fonction detect_anomalies() (ligne ~380 du fichier), modifiez les seuils selon vos normes :

```
# Seuil Z-score (défaut : 2.0)
if z > 2.0: # Changer selon votre tolérance

# Seuil écart budget (défaut : 8%)
if abs(row['ecart_ca_pct']) > 8: # Ex: 5% pour plus de sensibilité

# Seuil marge EBITDA (défaut : 6%)
if row['marge_ebitda'] < 6.0: # Ex: 10% pour un secteur à forte marge
```

5.4 Ajouter des Business Units

Pour ajouter des BU supplémentaires, modifiez generate_company_data() :

```
# Ajouter dans le DataFrame :
df['ca_bu_delta'] = (df['ca_reel'] * rng.uniform(0.08, 0.12,
n)).round(0).astype(int)

# Et dans generate_products_data() :
produits = ['Alpha', 'Beta', 'Gamma', 'Delta', 'Epsilon']
# Ajoutez les données correspondantes dans toutes les listes
```

5.5 Déploiement cloud

| Plateforme | Méthode | Coût | Documentation |
|-----------------------------|------------------------------|-----------|---------------------|
| Streamlit Community Cloud | Connecter dépôt GitHub | Gratuit | share.streamlit.io |
| Azure App Service | Docker ou déploiement direct | ~15€/mois | docs.microsoft.com |
| AWS EC2 / Elastic Beanstalk | Docker ou AMI | ~10€/mois | docs.aws.amazon.com |

| Plateforme | Méthode | Coût | Documentation |
|------------------|------------------|-------------|---|
| Google Cloud Run | Container Docker | Pay-per-use | cloud.google.com |

Dockerfile (déploiement conteneurisé)

```
FROM python:3.11-slim

WORKDIR /app
COPY requirements_pilote.txt .
RUN pip install --no-cache-dir -r requirements_pilote.txt

COPY pilote_cg.py .

EXPOSE 8501
CMD ["streamlit", "run", "pilote_cg.py",
      "--server.headless", "true",
      "--server.address", "0.0.0.0",
      "--server.port", "8501"]
```

6. RÉFÉRENCE TECHNIQUE

6.1 Architecture du code

| Section (lignes approx.) | Contenu |
|--------------------------|---|
| 1 – 35 | Docstring + imports (streamlit, plotly, pandas, numpy, sklearn, io, warnings) |
| 36 – 265 | Configuration page + Design System (dict COLORS + CSS injecté) |
| 265 – 460 | 5 fonctions de génération de données (@st.cache_data) |
| 460 – 565 | Chargement global des DataFrames + filtres de période/granularité |
| 565 – 615 | 3 helpers visuels : dark_layout(), kpi_card(), fmt_m() |
| 615 – 825 | Module 1 — Dashboard Exécutif |
| 826 – 972 | Module 2 — Budget vs Réel |
| 973 – 1101 | Module 3 — Forecasting ML |
| 1102 – 1207 | Module 4 — Détection d'Anomalies |
| 1208 – 1331 | Module 5 — Rentabilité |
| 1332 – 1479 | Module 6 — Trésorerie Prédictive |
| 1480 – 1795 | Module 7 — Import & Automatisation |
| 1796 – fin | Module 8 — Rapports Automatiques |

6.2 Fonctions clés — Référence

`generate_company_data(seed=42)`

Génère 24 mois de données financières synthétiques. Retourne un DataFrame avec : date, mois, ca_reel, ca_budget, charges_fixes, charges_variables, charges_personnel, ebitda_reel, ebitda_budget, ecart_ca, ecart_ca_pct, ecart_ebitda, marge_ebitda, charges_totales.

`detect_anomalies(df)`

Déetecte les anomalies selon 3 méthodes. Retourne un DataFrame avec colonnes : date, indicateur, valeur, z_score, severite (Critique/Modérée), type (Pic/Creux/Sur-performance/...), action (recommandation textuelle).

`generate_forecast(df, horizon=6)`

Prévoit le CA sur 'horizon' mois. Méthode : tendance linéaire (polyfit degré 1) + saisonnalité (moyenne mensuelle historique). Retourne un DataFrame avec date, mois, forecast, forecast_p10, forecast_p90.

`dark_layout(fig, height=380, title="")`

Applique le thème sombre unifié à n'importe quelle figure Plotly. Définit paper_bgcolor, plot_bgcolor, polices, couleurs des axes, marges. À appeler après toute création de figure.

fmt_m(v)

Formatte un nombre en notation compacte : $\geq 1M \rightarrow '1.23M\text{\euro}'$, $\geq 1k \rightarrow '123k\text{\euro}'$, sinon '123\text{\euro}'.

Utilisé pour tous les affichages de KPIs.

6.3 DataFrames et colonnes

| DataFrame | Lignes | Colonnes clés | Utilisé dans |
|--------------|------------|--|-------------------|
| df_main | 24 mois | ca_reel, ca_budget, ebitda_reel, marge_ebitda, charges_* | Modules 1,2,3,4,8 |
| df_12m | 12 mois | Sous-ensemble de df_main | Module 1 (défaut) |
| df_view | Variable | Filtré/agrégré selon filtres sidebar | Tous modules |
| df_prod | 6 produits | ca_ytd, marge_brute_pct, evolution_yoy, ecart_budget | Module 5 |
| df_clients | 10 clients | ca_ytd, dso_jours, marge_client, risque_recouvrement | Module 5 |
| df_cash | 90 jours | cash, cash_p10, cash_p90, encaissements, decaissements | Module 6 |
| df_anomalies | Variable | date, indicateur, severite, type, action | Modules 1,4,8 |
| df_forecast | 6 mois | forecast, forecast_p10, forecast_p90 | Modules 1,3 |

6.4 Gestion du cache Streamlit

Toutes les fonctions de génération/chargement de données sont décorées avec `@st.cache_data`. Cela signifie qu'elles ne sont exécutées qu'une seule fois au démarrage et que leurs résultats sont mis en cache pour toute la session. Avantage : navigation ultra-rapide entre les modules. Pour forcer le recalcul (par exemple après un changement de données), trois options :

- Appuyer sur 'C' puis Entrée dans l'interface (Clear cache)
- Redémarrer l'application (Ctrl+C dans le terminal puis streamlit run ...)
- Utiliser `st.cache_data.clear()` dans le code

7. RÉSOLUTION DE PROBLÈMES

7.1 Erreurs courantes et solutions

| Erreur | Cause probable | Solution |
|--|---|--|
| ModuleNotFoundError: No module named 'streamlit' | Dépendances non installées ou mauvais environnement virtuel | Vérifier que le venv est activé puis relancer pip install -r requirements_pilote.txt |
| SyntaxError: f-string expression part cannot include a backslash | Python 3.11 ou inférieur : backslash dans expression {} d'un f-string | Utiliser la version corrigée du fichier (variable pré-calculée avant le f-string) |
| TypeError: unsupported operand type(s) for +: 'int' and 'str' | add_vline() utilisé avec un axe X catégoriel (labels texte) | Remplacer add_vline par add_shape + add_annotation avec xref='x' |
| ValueError: could not convert string to float | Fichier CSV avec décimale ',' et séparateur ';' identiques | Utiliser sep=';' et decimal=',' pour les CSV français |
| MemoryError ou performance lente | Jeu de données trop volumineux sans pagination | Activer st.cache_data, réduire la période via le filtre, paginer les tableaux |

7.2 Questions fréquentes

Q : Comment utiliser mes propres données à la place des données démo ?

R : Allez dans le module Import & Automatisation, onglet 'Import Manuel'. Uploadez votre fichier CSV ou Excel en suivant le template téléchargeable. Ensuite, dans pilote_cg.py, remplacez l'appel generate_company_data() par votre propre fonction de chargement.

Q : Peut-on connecter plusieurs sources de données simultanément ?

R : Oui. Modifiez les fonctions de génération pour qu'elles agrègent plusieurs sources. Par exemple, charger le CA depuis l'ERP et le budget depuis un fichier Excel séparé, puis les fusionner avec pd.merge() sur la colonne date.

Q : Comment ajouter un nouvel indicateur au Dashboard ?

R : (1) Calculez l'indicateur dans generate_company_data() et ajoutez-le au DataFrame. (2) Dans le module Dashboard, ajoutez une colonne avec st.columns(6) à la place de st.columns(5). (3) Ajoutez le kpi_card() correspondant.

Q : L'application est-elle compatible avec Python 3.12 ?

R : Oui, PILOTE est compatible Python 3.11 et 3.12. La version 3.12 a levé la restriction sur les backslashes dans les f-strings, d'où l'absence d'erreur sur certains environnements. La version du fichier fournie est compatible avec les deux.

Q : Comment planifier l'exécution automatique du rapport ?

R : Pour une exécution planifiée sans interface, créez un script séparé report_auto.py qui importe les fonctions de pilote_cg.py et appelle send_alert() et la génération Excel. Planifiez ce script avec cron (Linux/macOS) ou le Planificateur de tâches (Windows).

```
# Exemple crontab (Linux) – rapport tous les jours à 7h  
0 7 * * * /chemin/venv/bin/python /chemin/report_auto.py  
  
# Rapport mensuel le 3 du mois à 8h  
0 8 3 * * /chemin/venv/bin/python /chemin/report_auto.py --type mensuel
```

8. FICHIER DE DONNÉES — pilote_donnees_financieres.csv

8.1 Présentation

Ce fichier CSV contient 36 mois de données financières synthétiques réalistes (janvier 2022 – décembre 2024), conçu pour être importé directement dans PILOTE sans modification.

Format : CSV · Séparateur : point-virgule (;) · Décimale : virgule (,) · Encodage : UTF-8 BOM

Dimensions : 36 lignes × 24 colonnes

Période : Janvier 2022 → Décembre 2024 (3 ans complets)

8.2 Dictionnaire de données

| Colonne | Type | Description | Exemple |
|-------------------|--------------------|--|------------|
| date | Date (YYYY-MM-DD) | Date de fin de mois | 2024-03-31 |
| mois_label | Texte | Label lisible du mois | Mar 2024 |
| annee | Entier | Année | 2024 |
| trimestre | Texte | Trimestre (T1 à T4) | T1 |
| jours_ouvertes | Entier | Nb jours ouvertes du mois | 21 |
| ca_reel | Entier (€) | Chiffre d'affaires réalisé | 4 250 000 |
| ca_budget | Entier (€) | Chiffre d'affaires budgétaire | 4 100 000 |
| ecart_ca_eur | Entier (€) | Écart CA réel - budget (peut être négatif) | +150 000 |
| ecart_ca_pct | Décimal (%) | Écart en pourcentage du budget | 3,66 |
| charges_fixes | Entier (€) | Loyer, amortissements, abonnements | 1 100 000 |
| charges_variables | Entier (€) | Matières premières, commissions | 1 730 000 |
| charges_personnel | Entier (€) | Salaires bruts + charges patronales | 860 000 |
| charges_totales | Entier (€) | Somme des 3 lignes de charges | 3 690 000 |
| ebitda_reel | Entier (€) | CA réel - charges totales | 560 000 |
| ebitda_budget | Entier (€) | EBITDA budgétaire | 490 000 |
| ecart_ebitda_eur | Entier (€) | Écart EBITDA réel - budget | +70 000 |
| marge_ebitda_pct | Décimal (%) | EBITDA réel / CA réel × 100 | 13,18 |
| dso_jours | Entier (jours) | Délai moyen de paiement clients | 42 |
| encaissements | Entier (€) | Flux d'encaissement du mois | 2 100 000 |

| Colonne | Type | Description | Exemple |
|---------------------|------------|-------------------------------------|-----------|
| decaissements | Entier (€) | Flux de décaissement du mois | 2 950 000 |
| flux_net_tresorerie | Entier (€) | Encaissements - Décaissements | -850 000 |
| ca_bu_alpha | Entier (€) | CA de la Business Unit Alpha (~41%) | 1 740 000 |
| ca_bu_beta | Entier (€) | CA de la Business Unit Beta (~31%) | 1 320 000 |
| ca_bu_gamma | Entier (€) | CA de la Business Unit Gamma (~28%) | 1 190 000 |

8.3 Procédure d'import dans PILOTE

5. Lancer l'application : streamlit run pilote_cg.py
6. Aller dans le module 📦 Import & Automatisation
7. Cliquer sur l'onglet 'Import Manuel'
8. Sélectionner le format 'CSV'
9. Glisser-déposer pilote_donnees_financieres.csv
10. Vérifier : séparateur = ',', décimale = ','
11. Vérifier le rapport qualité (0 valeurs nulles attendu)
12. Mapper les colonnes vers les indicateurs PILOTE
13. Cliquer sur 'Valider le Mapping & Importer'

IMPORTANT — Données synthétiques

Les données de ce fichier sont générées algorithmiquement à des fins de démonstration. Elles intègrent une tendance haussière réaliste (+85% de CA sur 3 ans), une saisonnalité annuelle, du bruit aléatoire et des anomalies ponctuelles. Elles ne représentent aucune entreprise réelle.

9. GLOSSAIRE

| Terme | Définition |
|------------------------------|--|
| EBITDA | Earnings Before Interest, Taxes, Depreciation & Amortization. Résultat opérationnel avant déductions financières. |
| DSO | Days Sales Outstanding. Délai moyen de paiement des clients en jours. DSO = (Créances clients / CA) × 365. |
| BFR | Besoin en Fonds de Roulement. Capital nécessaire pour financer le décalage entre encaissements et décaissements. |
| MAPE | Mean Absolute Percentage Error. Erreur moyenne absolue en %. Mesure de précision des modèles de prévision. |
| Z-score | Nombre d'écart-types séparant une valeur de la moyenne. $Z = (x - \mu) / \sigma$. Utilisé pour détecter les outliers. |
| P10 / P50 / P90 | Percentiles. P10 = 10% de chances d'être en dessous. P50 = médiane. P90 = 90% de chances d'être en dessous. |
| Intervalle de confiance (IC) | Plage de valeurs dans laquelle la vraie valeur se trouve avec une probabilité donnée (80%, 90% ou 95%). |
| Feature Engineering | Création de variables dérivées à partir des données brutes pour améliorer les modèles ML (ex: lag-1, rolling mean). |
| Waterfall / Pont | Graphique en cascade qui décompose un écart entre une valeur initiale et une valeur finale en sous-contributions. |
| ETL | Extract, Transform, Load. Pipeline d'intégration de données depuis les sources vers le système analytique. |
| Drift | Dérive du modèle. Dégradation progressive de la précision d'un modèle ML due à l'évolution des données en production. |
| RevPAR | Revenue Per Available Room. Indicateur hôtelier = Taux d'occupation × Prix moyen par chambre. |
| OEE | Overall Equipment Effectiveness. Indicateur industriel de performance des équipements de production. |
| KPI | Key Performance Indicator. Indicateur clé de performance mesurant l'atteinte d'un objectif stratégique. |
| YTD | Year To Date. Valeur cumulée depuis le début de l'année en cours. |