

Mission 8 : Réaliser un dashboard & Veille technologique

Classification d'images e-commerce par Deep Learning

Adrien NORMAND

Janvier 2026

Plan de la Présentation



Partie 1 : Dashboard (10 min)

- Dataset & Architecture
- Parcours utilisateur
- Graphiques interactifs
- Accessibilité WCAG
- Démonstration live



Partie 2 : Veille (7 min)

- Algorithme PanCAN (2025)
- Vision Transformers
- Résultats comparatifs
- Interprétabilité
- Limites & Recommandations



PARTIE 1

Dashboard Interactif

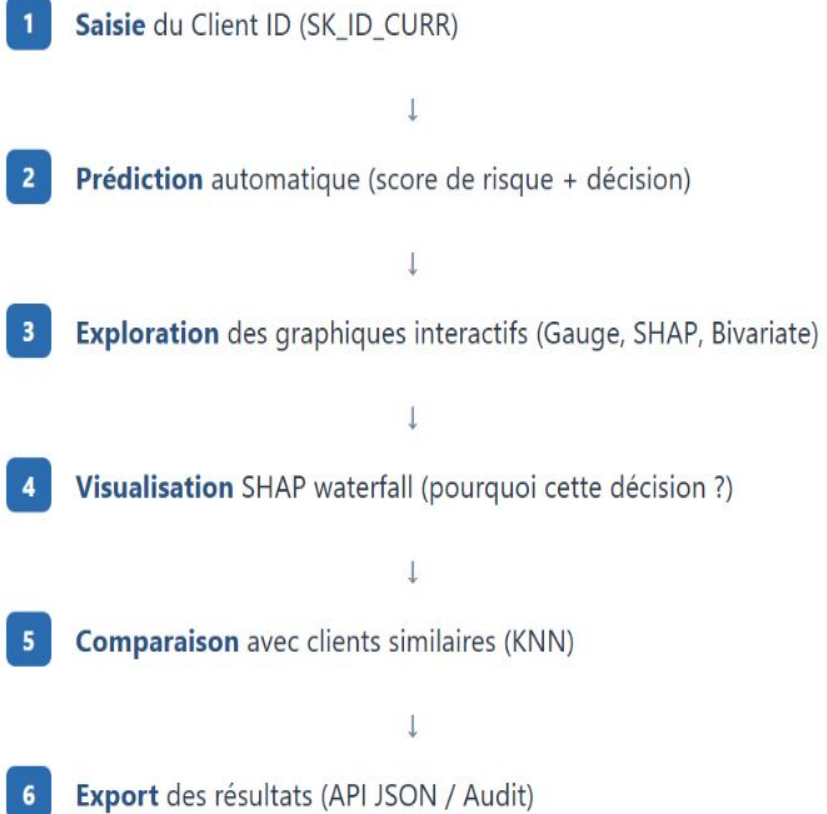


Stack Technique

- **Frontend** : Flask + Jinja2 Templates
- **Backend** : Flask + Gunicorn + LightGBM
- **Visualisation** : Plotly.js
- **Database** : PostgreSQL + SQLAlchemy
- **MLOps** : MLflow + Evidently
- **Cloud** : Docker + AWS Lightsail

Fonctionnalités

- Prédiction temps réel (API REST)
- Graphiques interactifs Plotly
- Explicabilité SHAP (RGPD Art. 22)
- Audit trail PostgreSQL
- Data Drift monitoring (Evidently)
- Documentation Swagger API



✓ 1.1.1 Contenu non textuel

→ Alt-text sur toutes les icônes SVG et graphiques Plotly

✓ 1.4.1 Utilisation de la couleur

→ Labels textuels + formes différentes (pas que couleur)

✓ 1.4.3 Contraste minimum (4.5:1)

→ Palette CSS variables validée avec contrast-checker

✓ 1.4.4 Redimensionnement texte (200%)

→ Layout responsive avec rem/em et media queries

✓ 2.4.1 Skip Navigation

→ Lien "Aller au contenu principal" sur chaque page

✓ 2.4.2 Titre de page

→ <title> descriptif unique sur chaque template

✓ 4.1.2 Nom, rôle, valeur

→ Attributs ARIA sur boutons, formulaires et navigation

Health Check

GET </api/health>

Client Data

GET </api/client/{id}>

Bivariate Analysis

GET </api/analysis/bivariate>

Audit Trail

GET </api/audit/predictions>

Prediction

POST </api/predict>

Similar Clients

GET </api/client/{id}/similar>

Model Info

GET </api/model/info>

Drift Report

GET </api/audit/drift-report>

Documentation Interactive

Swagger UI disponible à </api/docs>

Flasgger génère automatiquement la documentation OpenAPI à partir des docstrings.

DÉMONSTRATION EN DIRECT

1. Accès à l'URL déployée
2. Saisie d'un Client ID (ex: 333721, 110157, 268315)
3. Affichage de la prédiction + Score Gauge
4. Exploration des graphiques SHAP
5. Visualisation clients similaires
6. Consultation de l'audit trail

 **URL** : https://datascience-adventure.xyz/dashboard_mission7/

LightGBM

ALGORITHME

v13

VERSION MODÈLE

148K

CLIENTS DB

16

TESTS PYTEST

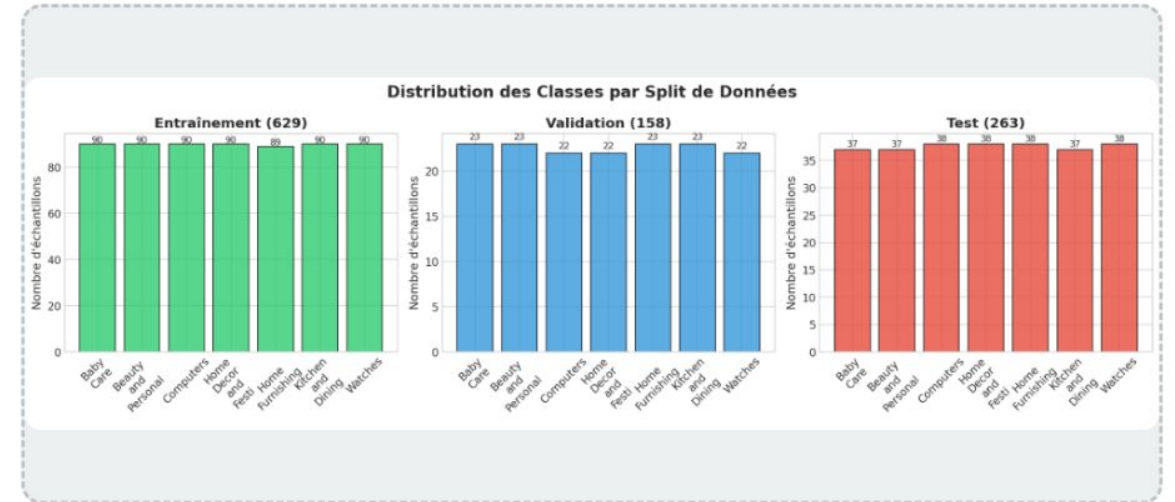


PARTIE 2

Veille Technologique

- **1 050 images** de produits
- **7 catégories** e-commerce
- Baby Care, Beauty, Computers
- Home Decor, Furnishing, Kitchen, Watches

Split	Images	Ratio
Train	629	60%
Validation	158	15%
Test	263	25%



Articles consultés (2025)

Auteurs	Titre	Source
Jiu et al.	PanCAN - Context Aggregation Networks	arXiv:2512.23486
Wang et al.	Vision Transformers Survey	Technologies Journal
Willis & Bakos	Multimodal Fusion Strategies	arXiv:2511.21889
Abulfaraj	Deep Ensemble Learning	BDCC Journal

 **Conférences de référence :** ICCV, NeurIPS, ICLR, CVPR

PanCAN = Panoptic Context Aggregation Network

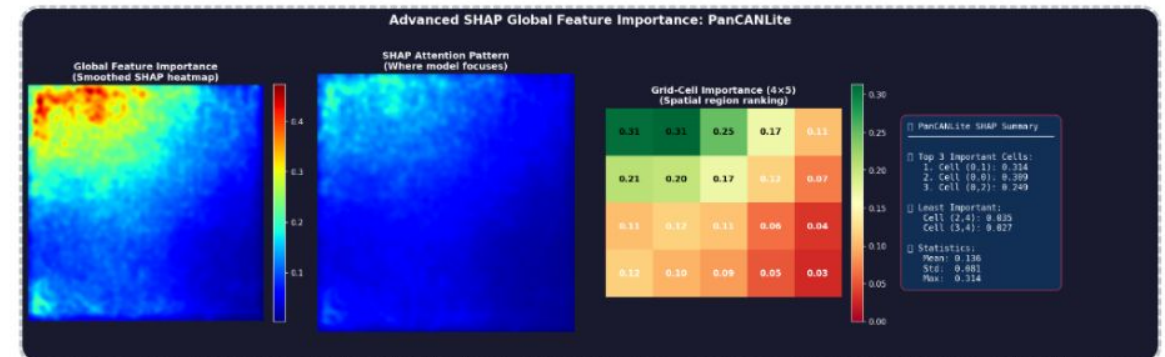
3 Piliers :

- 1 Grille multi-résolution
($8 \times 10 \rightarrow 4 \times 5 \rightarrow 2 \times 3 \rightarrow 1 \times 1$)
- 2 Agrégation contextuelle
(voisinage sur graphe)
- 3 Fusion multi-échelle

💡 Analogie simple :

Un CNN classique lit des **mots isolés**.

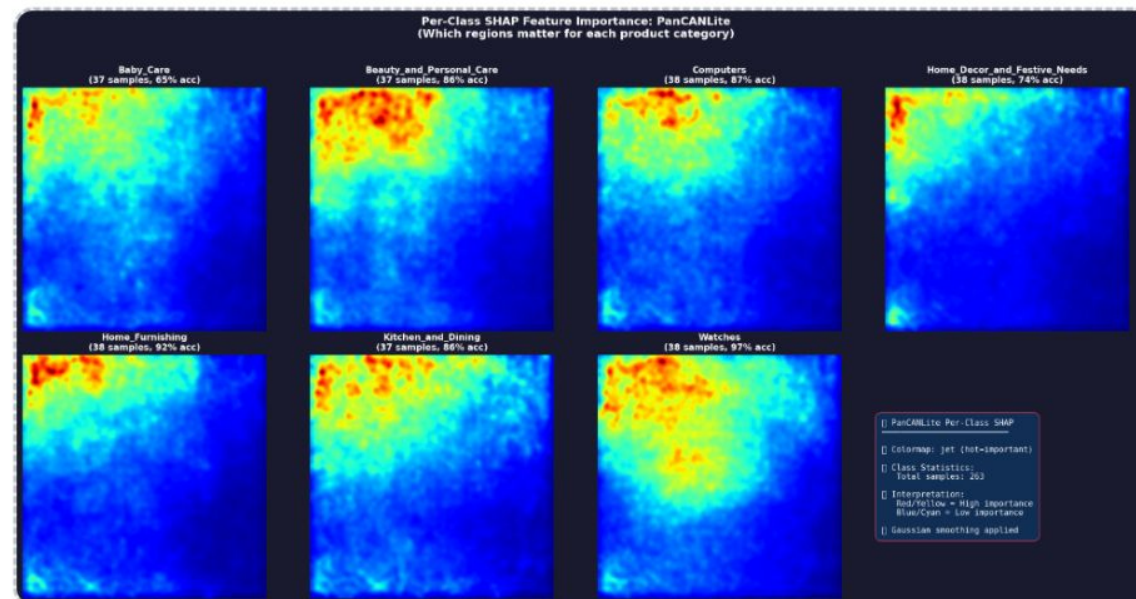
PanCAN lit la **phrase entière** et comprend que "casserole" + "four" = cuisine.



⚠ DÉCOUVERTE CRITIQUE

L'analyse SHAP révèle un **biais positionnel systématique**.

- Toutes les classes → importance **en haut à gauche**
- Pattern **NON discriminant** par catégorie
- Probable artefact du dataset Flipkart
- Risque : généralisation limitée



⚠ **Problème** : Dataset trop petit (629 images)
Ratio paramètres/données = **172,700:1** → Pertes NaN, non-convergence

Composant	PanCAN Original	PanCANLite
Backbone	ResNet-101	ResNet-50 (gelé)
Grilles	5 niveaux	1 niveau (4×5)
Ordres contextuels	3	2
Paramètres	108M	3.3M
Ratio params/data	172,700:1	5,226:1

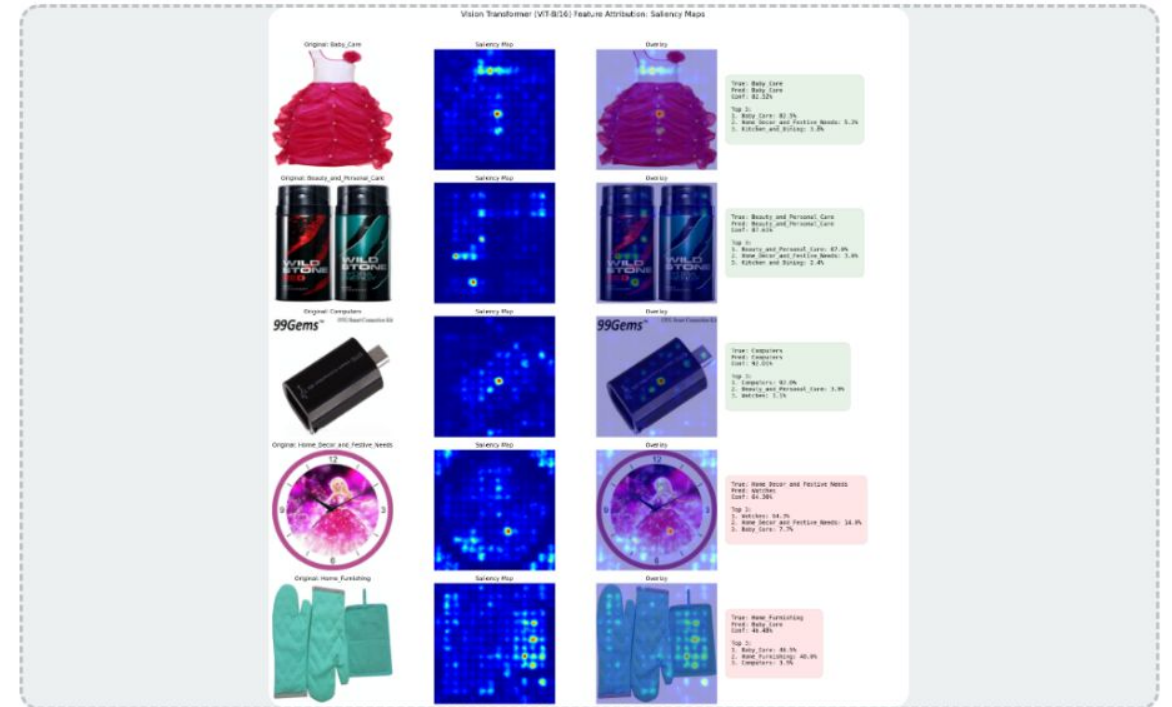
Constat : PanCANLite \approx VGG16 (84%)

→ Explorer les Transformers

Fonctionnement ViT :

1. Image \rightarrow 196 patches (16 \times 16)
2. Projection linéaire (768-dim)
3. Self-Attention globale
4. Token [CLS] \rightarrow Classification

Avantage clé : Relations longue distance dès la 1ère couche



Conclusion & Recommendations

Synthèse

- ✓ **Dashboard** : Déployé, accessible, interactif
- ✓ **Veille** : PanCAN testé, ViT validé
- ✓ **Meilleur modèle** : Multimodal (92.4%)
- ✓ **Biais identifié** : À valider sur data externe

Limites

- Biais spatial (SHAP haut-gauche)
- Dataset petit (1 050 images)
- Dépendance aux métadonnées texte

Recommandation Production


Modèle : Multimodal Fusion Lite
(EfficientNet-B0 + TF-IDF)

Accuracy : 92.40%

Latence : ~50ms/image

Fallback : ViT seul (87.83%)
si texte indisponible

✓ **Prêt pour la mise en production**
après validation sur dataset externe

Modèle	Accuracy	F1-Score	Paramètres
VGG16 (baseline)	84.79%	84.66%	107M
PanCANLite	84.03%	83.86%	3.3M
ViT-B/16	87.83%	87.61%	527K
Ensemble (3 modèles)	88.21%	88.07%	—
 Multimodal Fusion	92.40%	92.38%	658K

 **Progression** : +7.61 points vs baseline (84.79% → 92.40%)