

SOUTENANCE PORTFOLIO

ML DATA SCIENTIST

# William Le Roux

Machine Learning Data Scientist & Software Engineer

Epitech (Bachelor + Master IT) · UTT (Master InfoSec)

France · Français natif · Anglais C1

[GitHub — Septimus4](#)

[LinkedIn — william-le-roux](#)

Projet personnel : **LOCALTRIAGE** — Plateforme de triage support client par LLM local

# Méthodologie de construction du portfolio

## Démarche en 5 étapes

### 1. Inventaire

Recensement de 41 repositories GitHub (personnels, formation, OSS)

### 2. Sélection par pertinence

Filtrage sur les compétences ML/DS : classification, RAG, MLOps, deep learning, NLP

### 3. Structuration en 4 sections

Compétences · Réflexivité · Soft skills · Mind map

### 4. Format HTML responsive

Thème GitHub sombre, portfolio en ligne consultable

### 5. Validation croisée

Chaque compétence ↔ un projet ↔ des métriques mesurables

## Principe directeur

*Chaque affirmation est adossée à un livrable vérifiable : repo GitHub, rapport, métriques.*

## Structure du portfolio

### §1 Compétences & Projets

12 projets détaillés avec stack, métriques, liens GitHub

### §2 Capacité réflexive

Ce qui a changé, erreurs, leçons, évolution du regard

### §3 Soft skills

8 compétences illustrées par des exemples concrets

### §4 Mind map

Vue synthétique de l'ensemble du profil

# Profil & compétences techniques

## Machine Learning & Data Science



## Engineering & Infrastructure



## Langages de programmation



## Formation & communauté

**Epitech** — Bachelor + Master Informatique

**UTT** — Master Sécurité Informatique

Conférences : ETHDenver · ETHcc · Devcon · LeHack

OSS : Sponsor NumFocus, Django + 7 projets

# Panorama des projets réalisés

## Projets personnels

**LOCALTRIAGE**  
Triage support client · LLM local · RAG avec citations

**73.3%**  
routing accuracy

**Planar**  
LiDAR indoor scanning · Raspberry Pi · 195 tests

**Chronogen**  
Wordlist generator · PyPI · 16 stars · 6 forks

**Concord · Weebo**  
BERTopic + Neo4j · LLM workbench RTX 5090

## Projets de formation

**MLOPS**  
**Home Credit MLOps Pipeline**  
MLflow · Evidently · FastAPI · Docker · CI/CD

**CLASSIFICATION**  
**HR Analytics**  
Turnover prediction · Feature engineering · SHAP  
**+269%**  
F1-score

**RAG**  
**SportSee RAG Evaluation**  
RAGAS · NL→SQL · Logfire observability

**DEEP LEARNING**  
**Semi-Supervised MRI**  
ResNet-18 · Threshold calibration · Brain tumor

# LOCALTRIAGE — Contexte & analyse des besoins

## Problème métier

- **Triage manuel lent** : 15-30 min/shift sur le routage
- **Réponses incohérentes** : qualité variable entre agents
- **Silos de connaissance** : expertise dans les têtes
- **Contrainte vie privée** : pas de cloud AI (GPT-4 interdit)

## Parties prenantes

QUI	BESOIN	KPI
Agent support	Brouillons rapides	Taux acceptation
Team lead	Routage précis	% routage correct
Product manager	Insights produit	Détection thèmes
Sécurité / IT	Traitement local	Zéro fuite

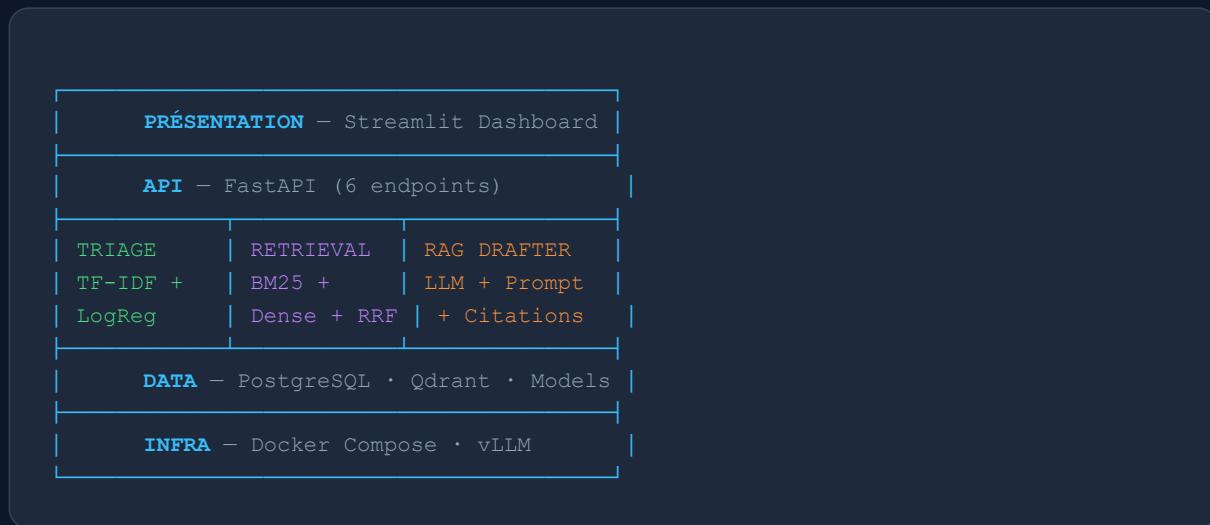
## KPIs cibles définis dans le BRD

KPI	BASELINE	CIBLE
Temps → 1er brouillon	8 min	<b>2 min</b>
Routage correct	30%	<b>90%</b>
Acceptation brouillon	N/A	<b>70%</b>
Hallucinations	N/A	< 2%
Latence p95	< 100ms	< 30s

## Documents produits



# LOCALTRIAGE — Architecture de la solution



## Pipeline de traitement



## Décisions techniques

CHOIX	DÉCISION
LLM	Qwen3:32B
Embeddings	BGE-Large-en
Vector DB	Qdrant
BDD	PostgreSQL
Retrieval	Hybride + RRF
Serving	Ollama / vLLM
API	FastAPI
UI	Streamlit

## Chaque choix documenté

Decision Matrix avec critères pondérés (qualité, vitesse, mémoire, licence, écosystème) et note composite pour chaque alternative.

# LOCALTRIAGE — Plan de projet (3 semaines)

## Semaine 1 — Fondations

- BRD, Context, architecture
- Schéma BDD, ingestion
- Classifier baseline (TF-IDF)
- Retrieval BM25

### ✓ MILESTONE 1

Système baseline opérationnel

## Semaine 2 — LLM & RAG

- Vector store + embeddings
- Retrieval hybride + RRF
- Intégration LLM
- Pipeline RAG + citations

### ✓ MILESTONE 2

Pipeline RAG opérationnel

## Semaine 3 — Eval & Deploy

- Evaluation harness
- API FastAPI
- Dashboard Streamlit
- Docker + documentation

### ✓ MILESTONE 3

Système production-ready

## Livrables produits

- ✓ Pipeline d'ingestion
- ✓ Audit baseline
- ✓ Routage LLM
- ✓ Drafting RAG
- ✓ Dashboard analytics
- ✓ Evaluation harness
- ✓ Déploiement Docker
- ✓ Documentation complète

## Chemin critique

Schema → Ingestion → Vector Store → Hybrid Retrieval  
→ RAG → API

**Parallélisation :** baselines développées en parallèle de l'intégration LLM.

**8/8**

livrables livrés

**3 sem.**

dans les délais

**84%**

test coverage

**Solo**

développement

# LOCALTRIAGE — Résultats mesurés

« On ne peut pas améliorer ce qu'on ne mesure pas. » — Approche baseline-first systématique.

**73.3%**

Routing accuracy  
+43.3pp vs baseline

**4.74/5**

Qualité brouillons  
+3.2 vs templates

**72%**

Taux d'acceptation  
cible 70% ✓

**24.3s**

Latence p95  
< 30s budget ✓

## Comparaison baseline → système

MÉTRIQUE	BASELINE	ACTUEL	Δ
Routing accuracy	30% (majorité)	73.3%	+43.3pp
Recall@5	70.8% (BM25)	91.7%	+20.8pp
Qualité drafts	1.5/5 (template)	4.74/5	+3.2
Citations	0%	94%	✓
Hallucinations	N/A	4.2%	< 5%
Détection P1	42% recall	89%	+47pp

## Détail qualité des brouillons

CRITÈRE	SCORE / 5
Correctness	4.5
Completeness	4.8
Tone / Clarity	4.7
Actionability	4.9
Citation Quality	4.8
<b>Moyenne</b>	<b>4.74</b>

# LOCALTRIAGE — Gestion des risques & pivots

## Risques matérialisés

<b>Librairie rank-bm25 trop lente</b> Pivot vers PostgreSQL Full-Text Search en 1 journée. <b>Leçon :</b> évaluer les choix de librairies contre les contraintes prod avant de s'engager.	<b>1 JOUR PERDU</b>
<b>Bottleneck d'annotation</b> Jeu d'évaluation réduit de 500 à 200 samples → intervalles de confiance plus larges. <b>Leçon :</b> démarrer l'annotation en parallèle dès le jour 1.	<b>JEU D'ÉVAL RÉDUIT</b>
<b>Scope creep dashboard</b> Demandes d'ajouts incessantes ("métriques de performance agent ?"). <b>Leçon :</b> phase gates strictes + définition de « done » explicite.	<b>RÉSISTÉ</b>

## Risques anticipés & mitigés

RISQUE	SCORE	MITIGATION
Modèle trop gros pour VRAM	6	Quantization INT8 + fallback 7B
Latence excessive	4	Budget/composant → p95 = 24.3s
Retrieval insuffisant	3	Hybride BM25 + Dense + RRF
Qualité données KB	4	Pipeline validation

## Ce que j'aurais fait différemment

- Commencer avec PostgreSQL FTS directement
- Dataset d'évaluation plus large dès le départ
- Architecture async dès le jour 1
- Caching de requêtes dans le MVP
- Demos stakeholders hebdomadaires

# Réflexivité — Ce que ce travail m'a apporté

## Avant la formation

- « Build first, evaluate later »
- Sauter directement au modèle
- Évaluation informelle : « ça a l'air bien »
- Documentation en afterthought
- Le modèle est fini quand il est entraîné

## Après la formation & les projets

- ✓ « Measure first, build with evidence »
- ✓ Baseline → gap analysis → solution
- ✓ Rubrics formelles (RAGAS, F1, confusion matrix)
- ✓ BRD, architecture, decision matrix, retrospective
- ✓ Le déploiement est le **DÉBUT** du cycle ML



# La formation en pratique — Apports concrets

## 4 transformations clés

### 1. Cycle ML structuré

Cadrage → Audit données → Baseline → Expérimentation → Déploiement → Monitoring

### 2. Rigueur statistique

Intervalles de confiance, cross-validation, splits train/val/test, métriques adaptées au problème

### 3. Traduction métier

Convertir F1/recall en KPIs business : 500K\$ économisés, 64% de détection, 72% d'acceptation

### 4. Mindset MLOps

Un modèle n'est pas fini tant qu'il n'est pas servi, monitoré et maintenable en production

## Évolution de ma perception du rôle

### Avant

Le Data Scientist est un **modélisateur** : entraîne des modèles, optimise des hyperparamètres, maximise l'accuracy dans un notebook.

### Après

Le Data Scientist est un **résolveur de problèmes data full-stack** :

- Gouvernance qualité des données
- Communication stakeholders
- Engineering de production
- Responsabilité éthique (vie privée)
- Boucles d'apprentissage continu

*Le déploiement est le **début** du cycle ML, pas la fin.*

# Synthèse — Compétences démontrées

COMPÉTENCE	PREUVE	Prochaines étapes
ML supervisé & classification	HR Analytics (+269% F1), LOCALTRIAGE (73.3%)	(LOCALTRIAGE Phase 2)
RAG & LLM	LOCALTRIAGE (4.74/5, 94% citations), RAG-Eval	→ Support multi-langue
MLOps & production	MLOps Pipeline (FastAPI + Evidently + CI/CD)	→ Ingestion temps réel
Deep learning	Semi-Supervised MRI (ResNet-18)	→ Pipeline de fine-tuning
NLP & topic modeling	Concord (BERTopic + Neo4j)	→ Classification multi-label
Infra LLM	Weebo (RTX 5090, quantization)	→ Dashboard performance agents
Gestion de projet	LOCALTRIAGE (3 sem, 8/8, solo, risk register)	
Documentation	BRD, architecture, decision matrix, retrospective	

## Merci

Questions & échanges

[github.com/Septimus4/LOCALTRIAGE](https://github.com/Septimus4/LOCALTRIAGE)

[Portfolio HTML en ligne](#)