

Annexe au CV : Portfolio de Projets

Ce document détaille les projets techniques que j'ai menés, en distinguant mon expérience professionnelle acquise en stage de mes réalisations académiques et personnelles.

Projets Professionnels (Stage chez Eddmon & Le Kompa)

Projets développés et mis en production lors de mon stage d'Ingénieur Machine Learning & IA (Avril - Août 2025) puis temps partiel.

Système de Transcription et d'Analyse d'Appels

Contexte	Les conseillers perdaient un temps considérable à retranscrire manuellement les appels clients dans le CRM. Le processus était source d'erreurs, d'oublis et manquait d'homogénéité.
Objectif	Automatiser la transcription des appels et générer des résumés structurés pour les intégrer directement au CRM, afin de libérer du temps pour les équipes et fiabiliser le suivi client.
Solution	Mise en place d'un pipeline automatisé :
Développée	<ul style="list-style-type: none">○ Capture et Transcription : Utilisation de l'API AssemblyAI pour la transcription des appels, choisie pour la haute qualité de sa diarisation (séparation des interlocuteurs).○ Analyse IA : Un modèle de langage enrichit la transcription avec les données du CRM (historique client, etc.), puis génère un résumé clair, les points clés de la demande et des propositions d'actions (création de ticket, relance, etc.).○ Intégration : Les résultats sont injectés directement dans le CRM via Laravel Cloud, rendant l'information accessible à toute l'équipe en temps réel.
Technologies	AssemblyAI , Python, Modèles de Langage (Mistral, GPT), Laravel Cloud , n8n.
Impact	Gain de 100% du temps de saisie manuelle pour les conseillers, amélioration de l'homogénéité et de la fiabilité des données dans le CRM, et réactivité accrue des équipes.

Création Automatique de Fiches et Demandes Clients

Contexte	La saisie manuelle des demandes clients dans le CRM était chronophage et sujette à des erreurs de retranscription. De plus, des propos bruts ou inadaptés pouvaient être transmis aux tuteurs, nuisant à la qualité de la communication.
Objectif	Automatiser la création de fiches clients structurées, claires et professionnelles à partir des transcriptions d'appels, en filtrant les informations non pertinentes.

Solution	Développement d'un pipeline de traitement avancé :
Développée	<ul style="list-style-type: none"> ○ Enrichissement Contextuel : Le système récupère la transcription de l'appel et l'enrichit avec le contexte du CRM (profil du client, historique). ○ Reformulation par IA : Un modèle de langage (GPT ou Mistral fine-tuné) reformule la demande pour en extraire l'essentiel. ○ Filtrage et Génération : L'IA filtre automatiquement les propos inadaptés et génère une fiche client structurée, qui est ensuite intégrée dans le CRM via Laravel Cloud.
Technologies	Python, Modèles de Langage (GPT, Mistral), RAG , GCP , Laravel Cloud .
Impact	Le temps de création d'une fiche client est passé de 3-5 minutes à quelques secondes . La qualité des données a été améliorée, réduisant les erreurs et professionnalisant les échanges avec les tuteurs.

Génération Automatique de Réponses aux SMS

Contexte	La gestion du volume élevé de SMS provenant des clients, prospects et tuteurs était une tâche répétitive qui entraînait des délais de réponse variables et une charge de travail importante pour les conseillers.
Objectif	Accélérer le traitement des SMS en fournissant aux conseillers des brouillons de réponses personnalisées et contextuelles, qu'ils n'ont plus qu'à valider.
Solution	Création d'un agent conversationnel :
Développée	<ul style="list-style-type: none"> ○ Extraction de Contexte : L'agent récupère le SMS entrant et le contexte associé depuis le CRM (historique des demandes, échanges précédents). ○ Génération de Réponse : Un modèle de langage (GPT fine-tuné ou Mistral) analyse le payload (SMS + contexte) et génère une réponse adaptée, en respectant le ton des conseillers. ○ Validation Humaine : La réponse est transmise comme brouillon au conseiller, qui peut la valider en un clic ou la modifier.
Technologies	Modèles de Langage (GPT, Mistral via Ollama), GCP , Python, API internes.
Impact	Le délai de réponse aux SMS a été divisé par 5 , améliorant significativement la réactivité et la satisfaction client, tout en réduisant le stress opérationnel des équipes.

Récapitulatif d'Appel et Génération de Mails Personnalisés

Contexte	La rédaction manuelle de mails de suivi après chaque appel (comprenant récapitulatif, devis, facture, etc.) était une tâche longue et fastidieuse, pouvant prendre jusqu'à 10 minutes par conseiller.
Objectif	Automatiser la génération de mails de suivi personnalisés, professionnels et cohérents avec la charte de l'entreprise.

Solution	Un pipeline complet de génération de contenu :
Développée	<ul style="list-style-type: none"> ○ Génération de Résumé : Le système utilise la transcription de l'appel enrichie des données CRM pour générer un résumé structuré. ○ Adaptation du Style : Grâce à un fine-tuning sur GCP et à une ingénierie de prompts avancée, l'IA adapte le ton et le style rédactionnel à celui du conseiller concerné. ○ Intégration en Brouillon : Le mail final est automatiquement intégré en tant que brouillon dans la boîte mail professionnelle du conseiller via l'API Gmail, permettant une validation humaine avant envoi.
Technologies	AssemblyAI , Python, Modèles de Langage (fine-tuning sur GCP), Gmail API .
Impact	Réduction drastique du temps de rédaction, passant de 10 minutes à quelques secondes. Amélioration de l'homogénéité et de la qualité des communications, renforçant l'image de marque de l'entreprise.

Algorithme de Matching pour Demandes "Niches"

Contexte	Les demandes très spécifiques (ex: tuteur de mathématiques parlant italien) nécessitaient une recherche manuelle de plusieurs heures dans une base de plus de 4000 tuteurs, avec un faible taux de succès.
Objectif	Développer un algorithme capable d'identifier et de classer rapidement les profils de tuteurs les plus pertinents pour les demandes complexes.
Solution	<ul style="list-style-type: none"> ○ Entraînement du Modèle : Un modèle de Machine Learning supervisé a été entraîné sur l'historique des attributions réussies, en utilisant des critères multiples (compétences, disponibilités, localisation, feedbacks clients, etc.). ○ Développement du Scoring : Le modèle génère un score de pertinence pour chaque tuteur actif, produisant une liste restreinte des 10 meilleurs profils pour chaque demande. ○ Déploiement et Interface : L'algorithme a été déployé via une API sur DigitalOcean et intégré à une interface simple permettant aux conseillers de visualiser les suggestions et de valider le choix final.
Développée	
Technologies	Scikit-learn , Python, SQL , DigitalOcean .
Impact	Le temps de recherche est passé de plus d'une heure à quelques secondes. Le taux de satisfaction des demandes "niches" a été significativement amélioré, transformant une contrainte en avantage compétitif.

Agent IA "Julie" - Assistante RH et Référente Tuteurs

Contexte	Le pôle RH faisait face à une charge administrative considérable pour le recrutement (gestion des dossiers, organisation des entretiens) et le suivi des tuteurs (réponse aux questions récurrentes).
-----------------	---

Objectif	Créer un agent IA polyvalent pour automatiser les tâches RH répétitives et servir de premier point de contact pour les tuteurs.
Développée	<ul style="list-style-type: none"> ○ Base de Connaissances et RAG : Une base documentaire interne (procédures RH, guides) a été créée. L'agent utilise le RAG (Retrieval-Augmented Generation) pour y puiser des informations et fournir des réponses précises. ○ Connexion aux Outils Métiers : "Julie" a été connectée à Gmail pour gérer les mails, à Calendly pour organiser les entretiens, et au CRM pour mettre à jour les dossiers. ○ Double Rôle : L'agent a été programmé pour endosser deux rôles : "Assistante RH" pour les candidats et "Référente Tuteurs" pour les intervenants actifs.
Technologies	LangChain, RAG, Vector Databases, Python, Gmail API, Calendly API.
Impact	Réduction majeure de la charge administrative du pôle RH, amélioration de l'expérience candidat grâce à des réponses rapides et personnalisées, et communication fluidifiée avec les tuteurs.

Agent de Prospection B2B pour les Écoles

Contexte	La prospection manuelle d'établissements scolaires pour la filiale Le Kompa était extrêmement chronophage, limitant le volume de contacts et ralentissant la croissance.
Objectif	Automatiser la prospection en générant des emails personnalisés et pertinents à grande échelle.
Développée	<ul style="list-style-type: none"> ○ Exploration Web (Scraping) : Un module de scraping analyse les sites web des établissements ciblés pour identifier les activités périscolaires déjà offertes. ○ Génération de Prompt Enrichi : Les informations collectées sont structurées et combinées avec le catalogue de services du Kompa pour créer un prompt très détaillé. ○ Rédaction par IA et Intégration : Un modèle de langage rédige un email de prospection sur mesure, mettant en avant les services complémentaires. Le mail est ensuite envoyé en brouillon dans la boîte des commerciaux via l'API Gmail.
Technologies	Python (BeautifulSoup, Scrapy), Modèles de Langage, Gmail API .
Impact	A permis de démarcher un volume beaucoup plus important d'écoles avec un haut niveau de personnalisation. Plus de 10 écoles ont été signées grâce au premier contact initié par cet agent, contribuant directement à la rentabilité du Kompa dès sa première année.

Projets Académiques et Personnels

Sélection de projets réalisés avant et pendant la maîtrise, démontrant une expertise en data science, machine learning, deep learning et GenAI, en lien direct avec l'analyse de données, la modélisation prédictive et la réduction des risques.

Prévision de Séries Temporelles Financières (LSTM/Transformer)

Contexte	Besoin de modéliser l'évolution de rendements d'indices et d'actions pour explorer des scénarios de risque de marché et de stress.
Objectif	Construire un modèle de prévision de séries temporelles capable de mieux capter les dynamiques non linéaires qu'un modèle classique (ARIMA, régression).
Solution Développée	<ul style="list-style-type: none">○ Constitution d'un dataset de plusieurs années de données financières (indices, actions), feature engineering (retours log, volatilité, moyennes mobiles).○ Entraînement et comparaison de modèles LSTM et Transformer (PyTorch) vs baseline ARIMA.
Technologies	Python, PyTorch , Pandas , scikit-learn , Matplotlib, Plotly.
Résultats	Réduction de la MAE/RMSE d'environ 40% par rapport à la baseline, meilleure capture des chocs de volatilité et amélioration de la qualité des scénarios de stress test.

Classification de Documents Financiers (Risque & Conformité) avec BERT

Contexte	Les institutions financières manipulent un grand volume de documents (rapports, mails, notes internes) qu'il est coûteux de catégoriser manuellement (risque opérationnel, conformité, client, etc.).
Objectif	Automatiser la classification de documents texte en plusieurs catégories liées au risque et à la conformité pour accélérer l'analyse et prioriser les cas critiques.
Solution Développée	<ul style="list-style-type: none">○ Constitution d'un corpus de textes annotés, prétraitement (tokenisation, nettoyage).○ Fine-tuning d'un modèle BERT (Hugging Face Transformers) pour classification multi-étiquettes.
Technologies	Python, Transformers (BERT), Hugging Face , scikit-learn, Matplotlib/Seaborn.
Résultats	F1-score macro ≈ 0,87 sur ~10k documents ; réduction significative du temps de tri manuel et meilleure détection des contenus sensibles (risque/compliance).

Pipeline ETL et Analyse Exploratoire Big Data (Spark & Pandas)

Contexte	Traitements de grands volumes de données transactionnelles hétérogènes pour préparer des cas d'usage d'analytique (détection d'anomalies, scoring, tableaux de bord).
Objectif	Mettre en place un pipeline ETL robuste et scalable, du brut jusqu'au dataset prêt pour la modélisation et la visualisation.

Solution Développée	<ul style="list-style-type: none"> ○ Ingestion de fichiers volumineux (> 500 Mo) avec Apache Spark, normalisation des schémas, jointures, agrégations. ○ Nettoyage et feature engineering avec Pandas, EDA et visualisation interactive (Plotly) pour mettre en évidence tendances et anomalies.
Technologies	Apache Spark, Pandas, SQL, Plotly, Seaborn.
Résultats	Temps de traitement divisés par un facteur 3 par rapport à un flux purement local ; datasets prêts pour des modèles de détection d'anomalies et de scoring de risque opérationnel.

⚙️ Prédiction Boursière par Reinforcement Learning	Développement d'un algorithme de Reinforcement Learning pour analyser et prédire les cours des actions. <i>Outils: Python, Keras, TensorFlow, Scikit-learn.</i>	prédition des cours du NASDAQ, intégrant des indicateurs financiers (RSI, MACD). <i>Outils: PySpark, Scikit-learn, Matplotlib.</i>
⚙️ Analyse de Données Cinématographiques	Exploitation d'un dataset de films (1990-2010) pour extraire des insights sur les tendances de l'industrie. <i>Outils: Python, bibliothèques de data science.</i>	⚙️ Jeu d'Échecs avec IA Création d'un jeu d'échecs avec une interface en Java et une logique d'IA en Python. <i>Outils: Java, Python.</i>
⚙️ Gestion de Big Data et Data Warehouse	Mise en place de pipelines de traitement de données hétérogènes avec Neo4j pour le non structuré et MySQL pour un Data Warehouse (analyse OLAP). <i>Outils: PySpark, Neo4j, MySQL.</i>	⚙️ Tic Tac Toe (Morpion) avec IA Développement d'un jeu de morpion avec une interface en Java et un backend d'IA en Python. <i>Outils: Java, Python.</i>
⚙️ Analyse du Marché Immobilier Français	Analyse de données foncières pour classifier les tendances du marché par type de bien, localisation et prix. <i>Outils: PySpark, Matplotlib, Scikit-learn.</i>	⚙️ Application de Fitness et Nutrition Conception d'une application mobile iOS intégrant programmes de fitness et conseils nutritionnels. <i>Outils: SwiftUI.</i>
⚙️ Jeu Vidéo RPG avec Données de Santé	Conception d'un RPG où les données de santé (sport, sommeil) constituent le cœur du gameplay pour gamifier l'activité physique. <i>Outils: Python, SwiftUI (iOS), Unity (C#).</i>	⚙️ Jeu de Bataille Navale Réalisation d'un jeu de bataille navale en C pur, sans librairies externes (sauf stdio.h et stdlib.h). <i>Outils: C.</i>
⚙️ Application d'Analyse Boursière NASDAQ	Développement d'une application d'analyse et de	⚙️ Application de Visionnage de Films avec IA Application de lecture de films locaux avec une IA extrayant titre, année et résumé. <i>Outils: Java, Python.</i>
		⚙️ Dispositif de Reconnaissance Vocale Développement d'un module sur Arduino pour

la reconnaissance vocale, associant matériel et réseaux neuronaux.

Outils: Arduino (C), Python.

⚙️ Modification de LLM pour Usage Quotidien

dien

Adaptation de modèles de langage open-source pour automatiser des tâches sur ordinateur, dans une approche d'assistant personnel type "JARVIS".

Outils: Python, C.