

Rapport de projet informatique

Analyse et estimation des prix d'enchères par IA.

Projet réalisé pour le Mardi 6 janvier 2026

Membres du groupe



Rayyane AIT AYACHA

n°étudiant : 43013167



Amy CHEN LI KWANG

n°étudiant : 44009648

Le dépôt GitHub contenant l'intégralité de notre projet est accessible via le lien suivant :

https://github.com/RAYYANE234/Projet_IA_Encheres.git

Remerciements

Au nom de toute l'équipe du projet d'Analyse et estimation des prix d'enchères par IA, je tiens à vous exprimer nos plus sincères remerciements pour votre soutien, votre guidance et votre expertise tout au long de ce projet. Votre dévouement et votre encouragement ont été essentiels pour notre succès.

Votre capacité à motiver et à partager vos connaissances a grandement enrichi notre expérience d'apprentissage. Votre expertise nous a aidé à surmonter les difficultés avec confiance et détermination.

Nous tenons également à remercier chaleureusement tous les participants du projet pour leur collaboration, leur créativité et leur travail acharné. Chacun d'entre vous a apporté une contribution unique et précieuse, ce qui a permis de réaliser ce projet avec succès. En travaillant ensemble, nous avons accompli des choses remarquables et nous sommes fiers du résultat que nous avons atteint.

Ce projet restera pour nous une expérience enrichissante et formatrice. Nous vous remercions sincèrement pour votre engagement et votre soutien tout au long de ce travail, et pour l'opportunité qui nous a été donnée de le réaliser.

Table des matières

1. Introduction.....	4
1.1 Contexte du projet:.....	4
1.2 Objectifs:.....	4
1.3 Technologies utilisées:.....	5
2. Architecture générale du système.....	6
2.1 Vue d'ensemble:.....	6
2.2 Diagramme d'architecture:.....	7
2.3 Flux de données:.....	7
3. Développement du système.....	8
3.1 Script Python :.....	8
3.2 Workflow n8n :.....	9
3.3 Analyse IA :.....	10
4. Collaboration.....	11
4.1 Travail en équipe:.....	11
4.2 Partage du workflow :.....	11
4.3 Tests collaboratifs:.....	11
5. Difficultés rencontrées.....	12
5.1 Méthodes testées et raisons des changements.....	12
5.2 Solution finale retenue.....	14
5.3 Problèmes techniques.....	14
5.4 Limitations des outils.....	15
6. Solutions apportées.....	15
6.1 Contournements techniques.....	15
6.2 Améliorations du workflow.....	16
6.3 Automatisation du pipeline.....	16
7. Résultats obtenus.....	16
7.1 Fonctionnement final.....	17
7.2 Estimation IA.....	17
8. Conclusion.....	20
9. Perspectives.....	21
10. Bibliographie.....	22
11. Webographie.....	23
12. Annexes.....	24

1. Introduction

1.1 Contexte du projet:

Avec le développement rapide des plateformes de vente en ligne et des sites d'enchères, une grande quantité de données est générée quotidiennement autour des produits mis en vente, notamment concernant leurs caractéristiques, leur popularité et leur prix final. Ces données constituent une source d'information précieuse pour analyser les comportements du marché et comprendre les mécanismes qui influencent la formation des prix.

Dans ce contexte, l'analyse de données et les techniques d'intelligence artificielle offrent des outils efficaces pour exploiter ces informations de manière automatisée. En particulier, les méthodes d'apprentissage non supervisé, comme le clustering, permettent d'explorer la structure des données sans connaissance préalable des catégories, tandis que les méthodes d'apprentissage supervisé rendent possible la prédiction de valeurs numériques telles que le prix final d'un produit.

Ce projet s'inscrit donc dans un cadre de **data analysis et de machine learning**, appliqué au domaine des ventes aux enchères. Il vise à exploiter des données issues de produits mis en vente afin de mieux comprendre les similitudes entre les articles, d'identifier des groupes de produits homogènes et, à terme, d'estimer automatiquement le prix d'un produit à partir de ses caractéristiques.

1.2 Objectifs:

L'objectif principal de ce projet est de concevoir un système capable d'estimer le prix d'un produit vendu aux enchères à partir d'un ensemble de données décrivant des produits similaires. Pour atteindre cet objectif, le projet repose sur plusieurs étapes complémentaires, allant de la collecte et du traitement des données jusqu'à l'application de méthodes d'apprentissage automatique.

Un premier objectif consiste à structurer et préparer les données afin de les rendre exploitables par des algorithmes d'intelligence artificielle. Cette étape inclut le nettoyage des données, l'uniformisation des formats et la transformation des informations textuelles en variables numériques. Une fois les données préparées, un second objectif est d'appliquer des techniques de clustering afin de regrouper automatiquement les produits selon leurs caractéristiques communes, sans utiliser de catégories prédéfinies.

Enfin, le projet vise à exploiter ces données enrichies et structurées pour entraîner un modèle d'apprentissage supervisé capable de prédire le prix d'un produit. Cette estimation pourra s'appuyer sur les caractéristiques du produit, ainsi que sur les groupes identifiés lors de l'étape de clustering. L'objectif global est donc de proposer une approche cohérente combinant **analyse exploratoire, clustering et prédiction de prix**, illustrant l'ensemble de la chaîne de traitement des données dans un projet de machine learning.




1.3 Technologies utilisées:

Le projet s'appuie sur un ensemble d'outils complémentaires permettant d'automatiser la collecte, le traitement et l'analyse des données. Les principales technologies utilisées sont les suivantes :

- Docker:

Docker a été utilisé pour faciliter l'exécution et la gestion de certains composants du projet, notamment n8n. Grâce à la conteneurisation, il a été possible d'isoler l'environnement d'exécution, d'éviter les problèmes de compatibilité et de garantir un fonctionnement identique sur différentes machines. L'utilisation de Docker a simplifié le déploiement du workflow n8n, en permettant de lancer l'outil rapidement sans configuration complexe. Cette approche a également rendu le système plus stable et plus reproductible, ce qui a été particulièrement utile lors des phases de test et de collaboration entre les membres de l'équipe.



<input type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
<input type="checkbox"/>	● n8n-container	bf8dae77a2fd	n8nio/n8n:latest	5678:5678	0.17%	11 hours ago	  

- Python:

Python constitue la base du développement du projet. Il a été utilisé pour collecter les informations saisies par l'utilisateur, structurer les données sous forme de fichiers JSON, communiquer avec le workflow n8n via des requêtes HTTP, générer les fichiers CSV et déclencher automatiquement le module d'analyse basé sur l'intelligence artificielle. Sa simplicité, sa lisibilité et la richesse de ses bibliothèques en font un langage parfaitement adapté à la création de pipelines de traitement de données.



- n8n (Automatisation / Orchestration):

n8n est une plateforme d'automatisation open-source qui a joué un rôle central dans l'orchestration du scraping. Grâce à son interface visuelle et à ses nombreux nœuds intégrés, il a permis de créer un workflow capable de recevoir des données via un webhook, d'interroger plusieurs plateformes de vente en ligne, de nettoyer les résultats et de renvoyer un JSON structuré au script Python. Son approche low-code a facilité la mise en place d'un processus clair, modulaire et facilement modifiable.



- Jina AI:

Jina AI a été utilisée dans le projet comme solution de scraping automatisé pour certaines plateformes. Grâce à ses capacités d'extraction de contenu à partir de pages web, Jina AI a permis de récupérer efficacement les informations nécessaires sans avoir à développer manuellement des scripts de scraping complexes. Son fonctionnement basé sur des requêtes structurées et des modèles optimisés pour l'analyse de contenu en ligne a facilité l'obtention de données fiables, même sur des sites dont la structure change fréquemment.



- Mistral AI:

L'intelligence artificielle Mistral a été intégrée pour analyser les données issues du scraping et produire une estimation de prix cohérente. Grâce à son API, il a été possible d'envoyer les caractéristiques du produit ainsi que les annonces récupérées, puis d'obtenir une analyse textuelle détaillée incluant un prix estimé, une fourchette de valeurs et une justification. Cette technologie a apporté une dimension intelligente au projet en allant au-delà du simple traitement de données.



2. Architecture générale du système

2.1 Vue d'ensemble:

L'architecture générale du système repose sur une organisation modulaire, dans laquelle chaque composant remplit une fonction précise et complémentaire. L'objectif global est d'assurer la collecte d'informations sur un produit, l'extraction automatisée d'annonces similaires sur plusieurs plateformes de vente en ligne, puis l'analyse de ces données à l'aide d'un modèle d'intelligence artificielle.

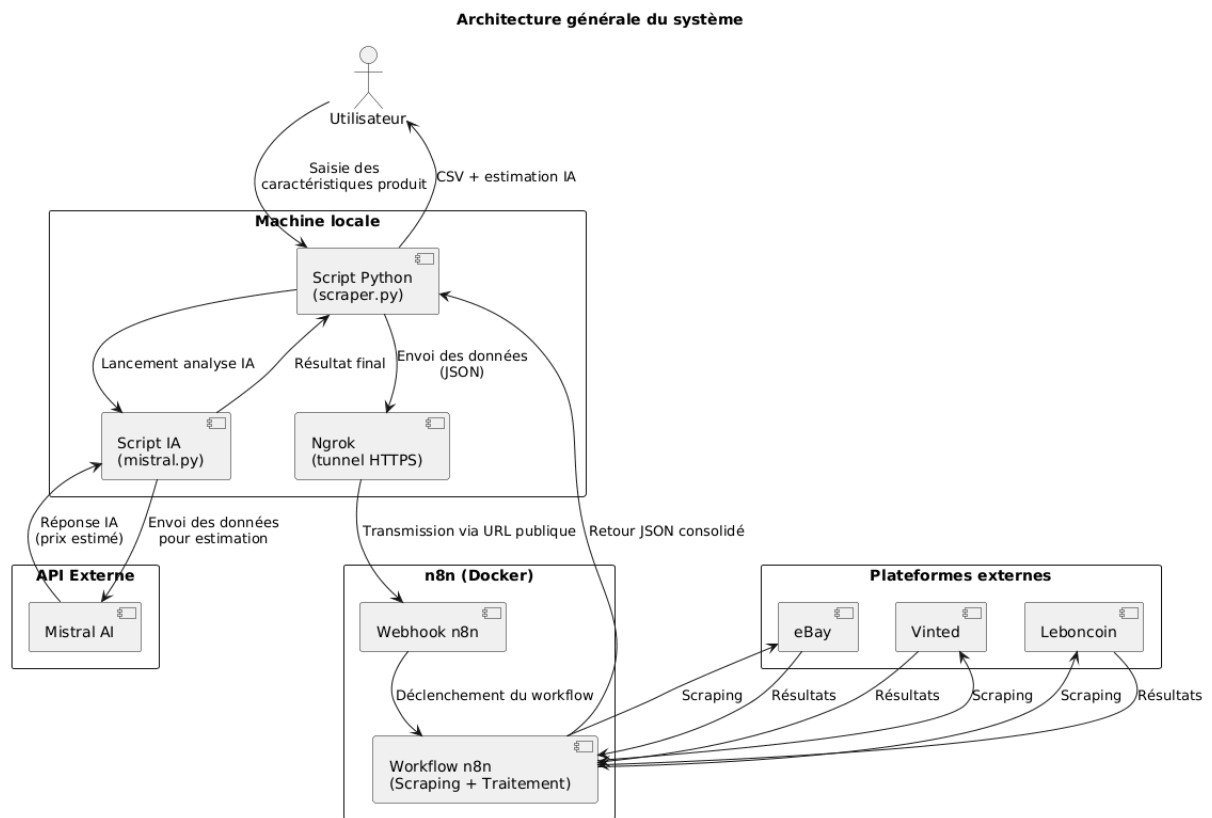
Le fonctionnement du système s'articule autour de plusieurs étapes clairement définies :

- **Script Python principal (point d'entrée du système)**
Ce script local constitue la première étape du processus. Il permet de recueillir les caractéristiques du produit fournies par l'utilisateur, de préparer les données nécessaires au traitement et de les transmettre au reste du système. Il assure ainsi l'initialisation et la cohérence des informations envoyées vers le workflow automatisé.
- **Workflow n8n (orchestration et scraping)**
Le workflow n8n joue un rôle central dans l'architecture. Il est chargé d'orchestrer le scraping des annonces sur différentes plateformes de vente en ligne, de traiter et filtrer les données récupérées, puis de produire un résultat structuré. Cette étape permet d'automatiser la collecte et le prétraitement des données issues de sources externes.
- **Script Python d'analyse basé sur l'API Mistral**
Un second script Python est dédié à l'analyse des données collectées. Il exploite l'API Mistral afin d'interpréter les informations issues du scraping et de générer une estimation de prix cohérente et argumentée. Ce composant constitue la phase finale du pipeline, orientée vers l'intelligence artificielle et l'aide à la décision.

La communication entre les différents composants est assurée par l'échange de fichiers au format **JSON** et par l'utilisation d'un **webhook** exposé via **ngrok**. Cette approche permet au système de fonctionner efficacement même lorsque le workflow n8n est exécuté en local.

Cette architecture distribuée présente **plusieurs avantages** : elle offre une grande flexibilité, facilite les phases de test et permet de modifier ou d'étendre chaque composant indépendamment des autres. L'ensemble forme ainsi un système complet, capable d'automatiser de manière cohérente la collecte, le traitement et l'analyse de données provenant de plusieurs sources externes.

2.2 Diagramme d'architecture:



2.3 Flux de données:

Le flux de données commence par la saisie des caractéristiques du produit dans le script Python, qui les transmet au workflow n8n. Ce dernier interroge les plateformes de vente, regroupe les annonces trouvées et renvoie un ensemble de données consolidé. Le script Python transforme ensuite ces résultats en fichiers CSV et lance l'analyse IA, qui produit une estimation finale. L'ensemble du flux est linéaire, automatisé et conçu pour assurer une circulation fluide des informations entre les différents modules.

3. Développement du système

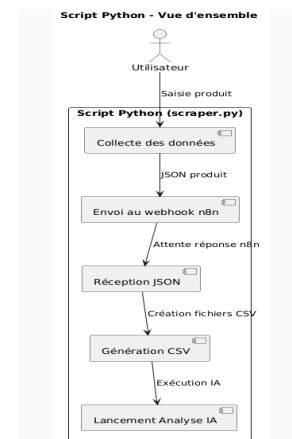
Le développement du système repose sur l'intégration cohérente de trois éléments principaux : un script Python chargé de piloter l'exécution globale, un workflow n8n dédié au scraping et au traitement des données, et un module d'analyse basé sur une intelligence artificielle. Ces composants ont été conçus de manière complémentaire afin de former un pipeline automatisé, depuis la collecte des informations jusqu'à la génération de l'estimation finale.

3.1 Script Python :

Le script Python constitue le point d'entrée du système et assure la coordination de l'ensemble du processus. Il recueille les informations saisies par l'utilisateur, les transmet au workflow n8n, récupère les résultats du scraping, génère les fichiers CSV et déclenche automatiquement l'analyse IA. Il joue donc un rôle central dans l'automatisation du pipeline.

- Etapes :


- Collecter les caractéristiques du produit auprès de l'utilisateur. Formate ces informations dans un fichier JSON.
- Envoie ce JSON au webhook n8n via une requête HTTP.
- Attend la réponse contenant les résultats du scraping.
- Reçoit un JSON consolidé avec les annonces trouvées.
- Génère plusieurs fichiers CSV (eBay, Vinted, Leboncoin).
- Lance automatiquement le script d'analyse IA.



La saisie des caractéristiques :

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS python + - [ ] [ ] ... ^ x

PS C:\Users\rayan\OneDrive\Bureau\Projet_encheres 2> python scraper.py
=== Création du JSON produit ===
Marque (ex: Zara)
> zara
Type de vêtement (ex: veste)
> veste
Matière (ex: laine mélangée)
> laine
Taille (ex: M)
> 
```

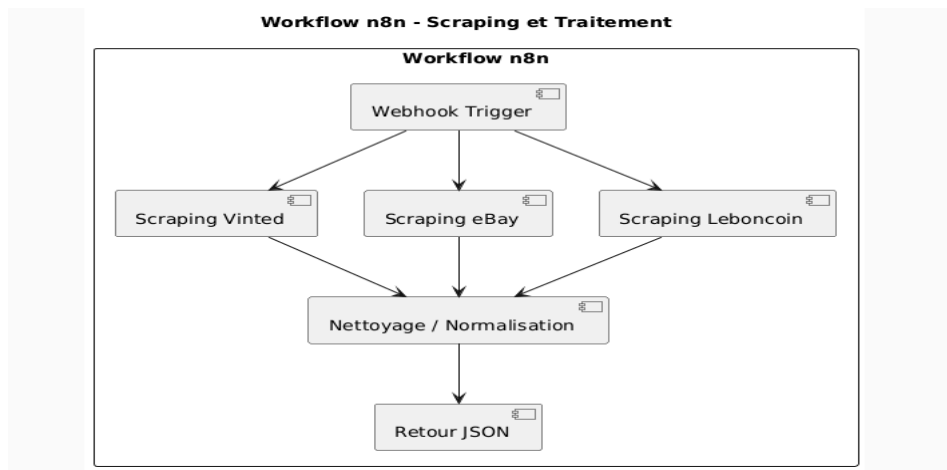


The screenshot shows a VS Code interface with a terminal window open. The terminal displays the output of a Python script that asks for clothing details. The user has provided the following input:

```
Type de vêtement (ex: veste)
> veste
Matière (ex: laine mélangée)
> laine
Taille (ex: M)
> S
Couleur (ex: beige)
> bleu
État (ex: très bon)
> neuf
```

The terminal status bar at the bottom indicates the file is at line 1, column 1, with 4 spaces, using UTF-8 encoding, CRLF line endings, and is a Python 3.9.13 file.

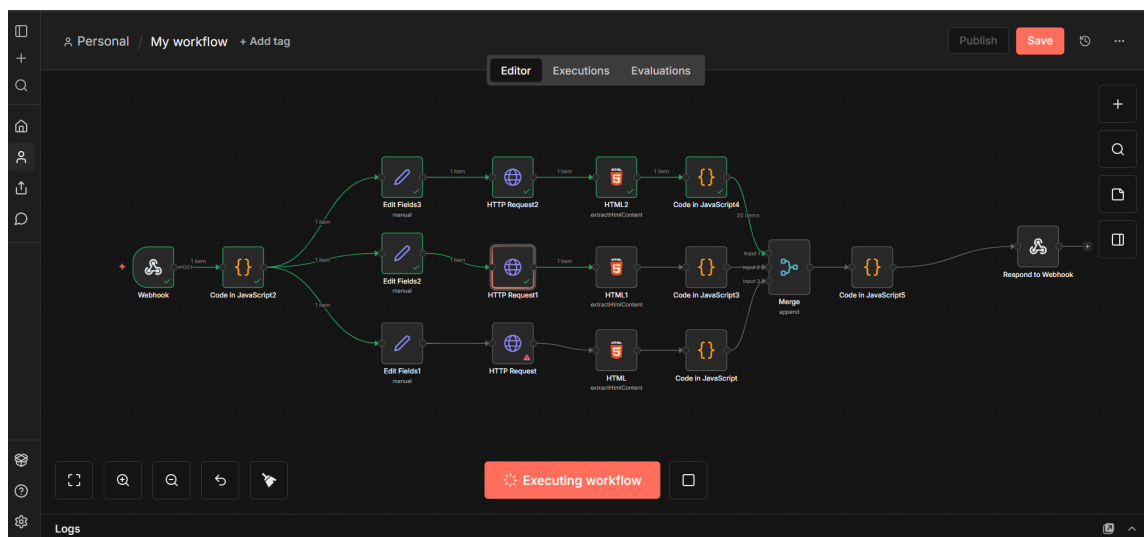
3.2 Workflow n8n :

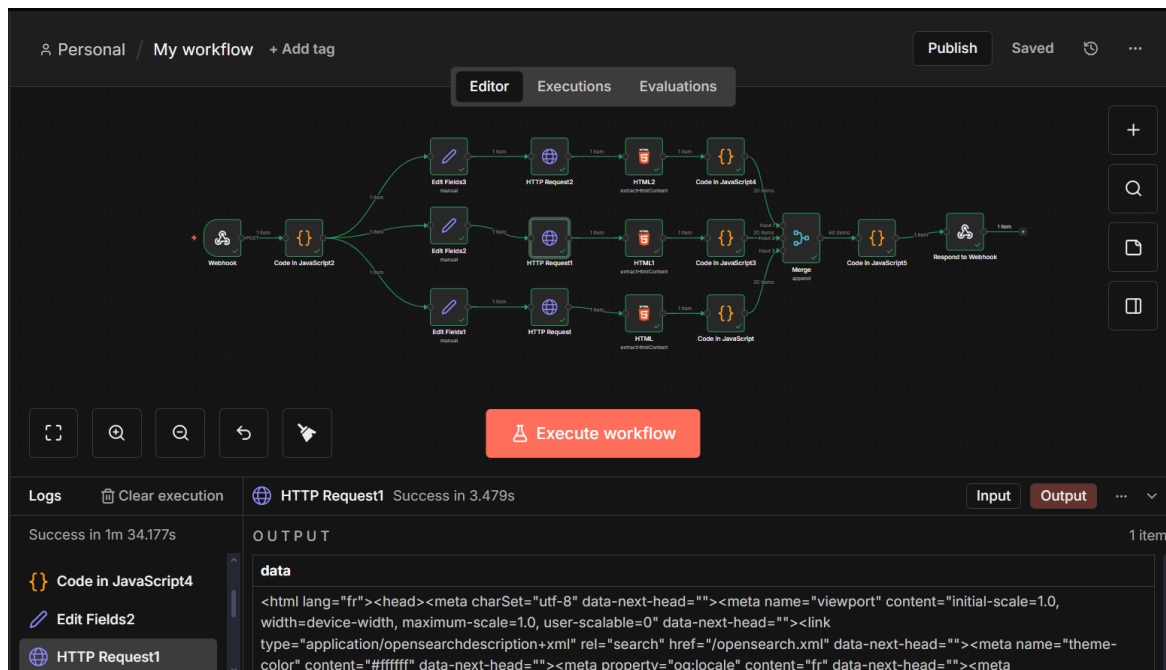


Le workflow n8n constitue le cœur de la partie scraping et traitement des données. À partir des informations reçues via le webhook, il interroge les différentes plateformes de vente, récupère les annonces pertinentes, les nettoie et les normalise avant de renvoyer un JSON consolidé au script Python. Il permet ainsi d'automatiser toute la phase de collecte d'informations.

- Etapes :

- Reçoit les données du produit via un webhook. Déclenche trois modules de scraping : Vinted, eBay, Leboncoin.
- Récupérer les annonces trouvées sur chaque plateforme.
- Nettoie les données (prix, titres, URLs, images).
- Normalise les champs pour avoir un format identique.
- Regroupe toutes les annonces dans un seul JSON.
- Renvoie ce JSON au script Python.





3.3 Analyse IA :

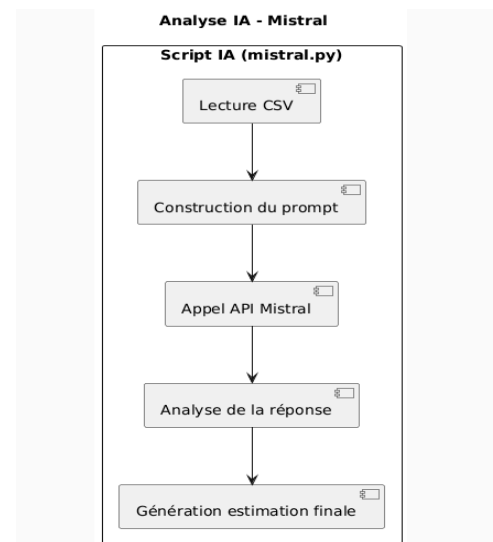
L'analyse IA intervient en fin de pipeline pour interpréter les données collectées et fournir une estimation de prix cohérente. À partir des fichiers CSV générés par le script Python, le module construit un prompt adapté, interroge l'API Mistral et produit une réponse contenant un prix conseillé accompagné d'une justification. Cette étape apporte une dimension intelligente au système.

- Etapes :

- Lit les fichiers CSV générés par le script Python. Analyser les données pour construire un prompt cohérent.
- Envoie ce prompt à l'API Mistral pour obtenir une estimation.
- Reçoit une réponse contenant un prix conseillé et une justification.
- Formate le résultat dans un fichier texte ou JSON.
- Renvoie l'estimation finale au script Python

4. Collaboration

La réalisation de ce projet s'est appuyée sur un travail collaboratif structuré, permettant de répartir les tâches, de tester les différentes étapes du pipeline et d'assurer une cohérence entre les composants développés. La collaboration a joué un rôle essentiel, notamment en raison de la nature distribuée du système, qui combine des scripts Python, un workflow n8n et des outils externes. Cette organisation a permis d'avancer efficacement tout en garantissant la qualité du résultat final.



4.1 Travail en équipe:

Le travail en équipe s'est déroulé de manière progressive et coordonnée. Chaque membre a pris en charge une partie spécifique du projet, tout en maintenant une communication régulière pour assurer l'intégration correcte des différents modules. L'un s'est concentré sur le développement du script Python et la gestion des données, tandis que l'autre a travaillé sur la configuration du workflow n8n et la mise en place du scraping. Cette répartition claire des responsabilités a permis de gagner du temps et d'éviter les conflits techniques. Les échanges fréquents ont facilité la résolution rapide des problèmes rencontrés et ont contribué à maintenir une vision commune du fonctionnement global du système.

4.2 Partage du workflow :

Le partage du workflow n8n a représenté un aspect important de la collaboration. Comme la version gratuite de n8n ne permet pas le partage direct d'un espace de travail, l'équipe a utilisé l'exportation et **l'importation de fichiers JSON** pour transférer le workflow d'un poste à l'autre. Cette méthode a permis à chaque membre de tester, modifier ou améliorer le pipeline sans dépendre d'un accès commun. Le workflow a ainsi pu être reproduit fidèlement sur plusieurs machines, garantissant une cohérence dans les tests et dans l'évolution du projet.

4.3 Tests collaboratifs:

Les tests collaboratifs ont été facilités par l'utilisation de ngrok, qui a permis d'exposer le webhook n8n sur une URL publique. Grâce à cela, le script Python développé par un membre de l'équipe pouvait interagir directement avec le workflow configuré par l'autre, même lorsque les deux travaillaient sur des environnements différents. Cette approche a permis de valider l'ensemble du pipeline dans des conditions proches de l'utilisation réelle. Les tests ont également servi à identifier des incohérences, à ajuster les formats de données échangées et à améliorer la robustesse du système. Cette phase de validation commune a été essentielle pour garantir le bon fonctionnement du projet dans son ensemble.



5. Difficultés rencontrées

Avant d'aboutir à la solution finale retenue pour ce projet, plusieurs méthodes ont été explorées afin de récupérer automatiquement des données issues de plateformes de vente et d'enchères en ligne. Ces tentatives ont permis d'identifier de nombreuses contraintes techniques et structurelles, qui ont conduit à l'abandon progressif de ces approches.

5.1 Méthodes testées et raisons des changements

a. Scraping direct d'eBay (Selenium / requests)

Objectif :

Récupérer automatiquement des données d'enchères de vêtements depuis la plateforme eBay.

Méthodes utilisées :

- Selenium pour simuler un navigateur web
- Tentatives complémentaires avec `requests` et `BeautifulSoup`

Problèmes rencontrés :

- Blocage automatique des requêtes par eBay
- Apparition fréquente de captchas
- Pages retournées vides ou contenant des résultats incomplets
- Données instables selon les requêtes et les sessions

Conclusion :

eBay dispose de protections anti-scraping avancées rendant l'extraction automatique peu fiable.

Méthode abandonnée, car non adaptée à une collecte stable et reproductible.

b. Scraping de sites d'enchères alternatifs (Catawiki)

Objectif :

Identifier une alternative à eBay proposant des enchères réelles exploitables.

Méthodes utilisées :

- Selenium
- Chargement dynamique des contenus via scroll automatique

Problèmes rencontrés :

- Chargement des annonces principalement via JavaScript
- Contenu partiellement invisible sans interactions complexes
- Données difficiles à structurer de manière cohérente
- Résultats incomplets et non reproductibles

Conclusion :

La structure fortement dynamique du site empêche une extraction fiable des données.

Méthode abandonnée pour manque de stabilité.

c. Utilisation d'APIs existantes (RapidAPI)

Objectif :

Accéder à des données d'enchères déjà structurées via des APIs publiques.

Méthodes utilisées :

- Recherche et tests d'APIs eBay et plateformes d'enchères sur RapidAPI

Problèmes rencontrés :

- APIs inactives, obsolètes ou non maintenues

- Accès limité ou nécessitant un abonnement payant
- Champs de données non personnalisables
- Quotas d'utilisation trop faibles

Conclusion :

Les APIs disponibles ne permettent pas la création d'un dataset adapté aux besoins du projet.

Méthode abandonnée pour manque de contrôle et de flexibilité.

d. Approche basée uniquement sur des mots-clés

Objectif :

Filtrer les annonces à partir de mots-clés simples (ex. « vêtement »).

Limites identifiées :

- Présence d'un bruit important dans les résultats
- Nombre élevé de faux positifs
- Précision insuffisante dans la sélection des annonces pertinentes

Conclusion :

Un filtrage basé uniquement sur des mots-clés ne permet pas une sélection fiable.

Nécessité d'un traitement intelligent, notamment via des techniques de NLP.

e. Tests avec Selenium et le navigateur Chrome

Objectif :

Automatiser la récupération des annonces sur le site ShopGoodwill à l'aide du navigateur Google Chrome.

Méthode utilisée :

- Utilisation de Selenium pour ouvrir automatiquement le navigateur
- Saisie du mot-clé dans la barre de recherche
- Tentative de récupération des annonces affichées

Problèmes rencontrés :

- Timeout lors du téléchargement automatique du driver Chrome
- Ouverture de pages blanches ou absence de résultats récupérés
- Difficulté à interagir avec les éléments générés dynamiquement via JavaScript

Conclusion :

L'automatisation avec Chrome s'est révélée instable et peu fiable.

Méthode abandonnée en raison des difficultés techniques persistantes.

f. Tests avec Selenium et le navigateur Microsoft Edge

Objectif :

Contourner les limitations rencontrées avec Chrome en utilisant un autre navigateur.

Méthode utilisée :

- Adaptation du script Selenium pour Microsoft Edge
- Spécification manuelle du chemin du driver (`msedgedriver.exe`)
- Configuration des options du navigateur pour réduire les messages d'automatisation
- Apparition de pop-ups de consentement bloquant l'accès à la barre de recherche
- Erreurs d'interaction avec les éléments de la page :
 - `ElementNotInteractableException`
 - `ElementClickInterceptedException`
- Absence totale d'annonces récupérées malgré une saisie correcte du mot-clé

Conclusion :

Le changement de navigateur n'a pas permis de résoudre les problèmes rencontrés.

Méthode abandonnée, Selenium restant inadapté pour ce site.

5.2 Solution finale retenue

La réalisation de ce projet avec la nouvelle solution trouvée a mis en évidence plusieurs difficultés liées à la fois aux aspects techniques et aux limites des outils utilisés. Certaines étapes ont nécessité des ajustements importants, notamment lors de l'intégration des différents modules, de la communication entre les composants et de la gestion des données issues du scraping. Ces obstacles ont contribué à affiner l'architecture du système et à renforcer la compréhension des contraintes liées à l'automatisation et à l'analyse de données.

5.3 Problèmes techniques

Plusieurs problèmes techniques ont été rencontrés au cours du développement.

- La communication entre le script Python et le webhook n8n a parfois été instable, notamment lorsque ngrok générait une nouvelle URL après chaque redémarrage. Cela nécessitait une mise à jour manuelle dans le script, ce qui ralentissait les tests.
- Le scraping des plateformes présente des difficultés liées aux variations de structure des pages, aux protections anti-bots et aux données parfois incomplètes ou incohérentes. Ces variations obligent à adapter régulièrement les requêtes ou les méthodes d'extraction.
- La gestion des formats JSON et CSV a également demandé une attention particulière, car les données renvoyées par les plateformes n'étaient pas toujours homogènes. Il a fallu normaliser les champs pour éviter des erreurs lors de l'analyse IA.
- L'intégration entre les deux scripts Python (scraping et IA) a nécessité une synchronisation précise, notamment pour s'assurer que les fichiers CSV étaient correctement générés avant d'être analysés.

5.4 Limitations des outils

Certaines limites des outils utilisés ont également influencé le déroulement du projet.

- La version gratuite de n8n ne permet pas le partage direct d'un espace de travail, ce qui a compliqué la collaboration. Il a fallu passer par l'exportation et l'importation manuelle du workflow pour travailler à plusieurs.
- Ngrok, dans sa version gratuite, génère une nouvelle URL à chaque lancement, ce qui empêche une automatisation complète et impose une reconfiguration régulière du script Python.
- Jina AI, utilisé pour le scraping, présente des limites lorsque les plateformes modifient leur structure ou mettent en place des protections renforcées. Cela peut entraîner des résultats incomplets ou des erreurs d'extraction.
- L'API Mistral, bien que performante, dépend fortement de la qualité des données fournies. Si les annonces récupérées sont trop hétérogènes ou peu nombreuses, l'estimation produite peut manquer de précision.
- Enfin, Docker facilite l'exécution de n8n, mais nécessite une configuration correcte du réseau et des volumes. Une mauvaise configuration peut empêcher le workflow de fonctionner ou entraîner une perte de données.

6. Solutions apportées

La mise en place du système a nécessité plusieurs ajustements pour contourner les difficultés rencontrées et optimiser le fonctionnement global du pipeline. Les solutions apportées ont permis d'améliorer la stabilité du scraping, de renforcer la cohérence des données échangées et de fluidifier la communication entre les différents modules. Ces améliorations ont contribué à rendre le système plus robuste, plus automatisé et plus facile à maintenir.

6.1 Contournements techniques

Plusieurs solutions ont été mises en place pour résoudre les problèmes techniques rencontrés au cours du développement.

- Pour pallier les changements fréquents d'URL générés par ngrok, une variable de configuration externe a été introduite afin de modifier rapidement l'adresse du webhook sans toucher au code.
- Les variations de structure des pages sur Vinted, eBay et Leboncoin ont été gérées en adaptant les requêtes et en ajoutant des vérifications supplémentaires pour éviter les erreurs lorsque certains champs étaient absents.
- La normalisation des données a été renforcée en imposant un format commun pour les prix, les titres et les URLs, ce qui a permis d'éviter des incohérences lors de l'analyse IA.
- Des blocs de gestion d'erreurs ont été ajoutés dans le script Python pour éviter les interruptions en cas de réponse incomplète ou de délai dépassé.

6.2 Améliorations du workflow

Le workflow n8n a été progressivement optimisé pour garantir un traitement plus fiable et plus structuré des données.

- Les nœuds de scraping ont été réorganisés pour exécuter les requêtes de manière indépendante, ce qui permet de continuer le traitement même si une plateforme ne répond pas.
- Un module de nettoyage centralisé a été ajouté afin de regrouper les opérations de filtrage, de normalisation et de validation des données.
- Le retour JSON envoyé au script Python a été simplifié pour ne contenir que les informations essentielles, facilitant ainsi la génération des fichiers CSV.
- Des logs internes ont été intégrés dans n8n pour faciliter le débogage et comprendre rapidement l'origine d'un problème lors des tests collaboratifs.

6.3 Automatisation du pipeline

L'automatisation du pipeline a été renforcée afin de réduire les interventions manuelles et d'assurer une exécution fluide du début à la fin.

- Le script Python a été conçu pour enchaîner automatiquement toutes les étapes : collecte des données, envoi au webhook, réception du JSON, génération des CSV et lancement de l'analyse IA.
- L'analyse IA a été intégrée comme une étape finale déclenchée automatiquement, garantissant que l'utilisateur obtient directement une estimation sans manipulation supplémentaire.
- L'utilisation de Docker pour exécuter n8n a permis de stabiliser l'environnement et d'éviter les problèmes liés aux dépendances ou aux configurations locales.
- L'ensemble du pipeline peut désormais être relancé rapidement, ce qui facilite les tests, les ajustements et les démonstrations.

7. Résultats obtenus

Le système développé atteint pleinement son objectif : automatiser la collecte d'annonces sur plusieurs plateformes, structurer les données obtenues et produire une estimation de prix cohérente grâce à l'intelligence artificielle. Les différents modules — script Python, workflow n8n, scraping Jina AI et analyse Mistral — fonctionnent désormais de manière fluide et coordonnée. Les tests réalisés ont permis de valider la stabilité du pipeline, la qualité des données collectées et la pertinence des estimations générées.

7.1 Fonctionnement final

Le fonctionnement final du système se déroule de manière entièrement automatisée, depuis la saisie initiale jusqu'à la génération du résultat final.

- L'utilisateur saisit les caractéristiques du produit dans le script Python.
- Le script envoie ces informations au webhook n8n via ngrok.
- Le workflow n8n interroge automatiquement Vinted, eBay et Leboncoin grâce à Jina AI.

- Les annonces récupérées sont nettoyées, normalisées et regroupées dans un JSON unique.
- Le script Python génère des fichiers CSV pour chaque plateforme et un CSV global.
- Le module d'analyse IA lit ces données et interroge Mistral pour obtenir une estimation.
- Le système renvoie à l'utilisateur un prix conseillé accompagné d'une justification détaillée.

Ce fonctionnement séquentiel et automatisé garantit une exécution rapide, reproductible et fiable.

7.2 Estimation IA

L'estimation produite par l'intelligence artificielle repose directement sur les données collectées et normalisées. Le modèle Mistral analyse les tendances observées dans les prix, l'état des produits et les variations entre plateformes pour proposer une estimation réaliste.

- L'IA identifie les valeurs extrêmes (prix trop bas ou trop élevés) et les exclut de l'analyse.

```
=== Résultat IA ===
### **Estimation de prix pour la veste Zara en laine (neuf, taille S, bleu)**

#### **Résultats de l'analyse :**
- **Prix estimé :** **60-80 €**
- **Prix minimum conseillé :** **50 €** (pour une vente rapide)
- **Prix maximum possible :** **110 €** (si la veste est rare ou très demandée)

---
```

- Le modèle fournit une justification textuelle expliquant les critères retenus (état, accessoires, variations entre plateformes).

```
### **Justification :**
1. **État "neuf" :** Les vestes Zara en laine neuves se vendent généralement entre **70 € et 160 €** sur eBay (ex. : 110 €, 165 €). Cependant, sur Vinted et Leboncoin, les prix sont plus bas (20-60 € pour du neuf ou très bon état).
2. **Taille S** : Les petites tailles sont souvent plus recherchées, ce qui peut justifier un prix légèrement plus élevé.
3. **Couleur bleu** : Peu d'annonces similaires en bleu, mais les vestes en laine de couleur classique (bleu marine, gris) se vendent bien.
4. **Comparaison avec les annonces proches** :
  - **eBay** : Les vestes neuves en laine Zara (taille S) oscillent entre **75 € et 165 €**.
  - **Vinted** : Les vestes en très bon état ou neuves (taille S) se situent entre **20 € et 60 €**.
  - **Leboncoin** : Les vestes similaires sont proposées entre **10 € et 35 €**, mais souvent en occasion.

**→ Un prix entre 60 € et 80 € est réaliste pour une vente équilibrée (ni trop longue, ni sous-évaluée).**

---
```

- Cette étape présente les facteurs pris en compte dans l'estimation du prix.

```
2. **Facteurs influençant le prix :**
- **Couleur verte** : Peu d'annonces exactement vertes, mais les tons kaki/olive (proches) valent **15-40 €**.
- **Matière laine** : Les vestes en laine pure ou mélangée se vendent mieux que les synthétiques.
- **Taille S** : Très demandée, donc prix légèrement plus élevé qu'une taille M.
- **État "bon"** : Les vestes en "très bon état" ou "neuves sans étiquette" atteignent **40-70 €**, tandis que les "bon état" restent sous **30 €**.
```

- Le programme affiche également une liste d'annonces similaires issues de différentes plateformes (eBay, Vinted et Leboncoin), permettant de comparer les prix et les caractéristiques des articles afin d'aider l'utilisateur à estimer une valeur de référence.

```
### **Annonces les plus proches (référence) :**
1. **eBay** :
  - **MANTEAU ZARA NEUF 100% LAINE AVEC CEINTURE [...] BLEU** → **165,34 €** (neuf, taille non précisée)
  - **ZARA NAVY BLUE DOUBLE-BREASTED 100% WOOL KNIT JACKET SIZE M** → **110,22 €** (neuf, taille M mais proche)
  - **VESTE EN TRICOT ZARA BLEU MARINE [...] 100% LAINE TAILLE M** → **75,78 €** (neuf, taille M)
2. **Vinted** :
  - **Veste Zara** (neuf avec étiquette, taille S) → **20 €**
  - **Est en laine Zara** (neuf sans étiquette, taille S) → **35 €**
  - **Manteau Zara baby blue** (très bon état, taille L) → **27 €** (couleur proche)
3. **Leboncoin** :
  - **VESTE BLEU MARINE T36 FEMME** → **13 €** (occasion, taille S)
  - **MANTEAU CABAN LAINE BLEU MARINE** → **10 €** (occasion)
---
```

- Le programme fournit également des recommandations de vente, incluant le choix de la plateforme, des conseils de présentation de l'annonce et une estimation du prix de départ afin d'optimiser les chances de vente.

```
### **Recommandations pour la vente :**
- **Plateforme idéale** : **Vinted** (meilleur compromis prix/visibilité) ou **eBay** (si la veste est rare).
- **Photos** : Mettre en avant l'étiquette "neuf", la composition (100% laine) et les détails (boutons, coupe).
- **Description** : Préciser "neuf avec étiquette", la taille exacte, et les mots-clés comme "veste hiver femme", "laine chaude", "style élégant".
- **Prix de départ** : **75 €** (négociable à 60-65 € si besoin).

Si la veste a des détails uniques (doubleure, boutons originaux), le prix peut être ajusté à la hausse.
```

- ❖ **L'estimation IA constitue ainsi une aide à la décision fiable, capable de synthétiser automatiquement des données hétérogènes pour proposer un prix réaliste et argumenté. enfin, voici l'illustration du fonctionnement global du programme :**

```
EXPLORER    PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PROJET_ENCHERES 2
ebay.csv
estimation_mistral.txt
leboncoin.csv
mistral.py
produit.json
resultats.json
scraper.py
vinted.csv

OUTLINE
TIMELINE

Ln 1, Col 1  Spaces: 4  UTF-8  CRLF  Plain Text

=== Résultat IA ===
### **Estimation de prix pour la veste Zara en laine (neuf, taille S, bleu)**

### **Résultats de l'analyse :**
- **Prix estimé** : **60-80 €**
- **Prix minimum conseillé** : **50 €** (pour une vente rapide)
- **Prix maximum possible** : **110 €** (si la veste est rare ou très demandée)

---

### **Justification :**
1. **État "neuf"** : Les vestes Zara en laine neuves se vendent généralement entre **70 € et 160 €** sur eBay (ex. : 110 €, 165 €). Cependant, sur Vinted et Leboncoin, les prix sont plus bas (20-60 € pour du neuf ou très bon état).
2. **Taille S** : Les petites tailles sont souvent plus recherchées, ce qui peut justifier un prix légèrement plus élevé.
3. **Couleur bleu** : Peu d'annonces similaires en bleu, mais les vestes en laine de couleur classique (bleu marine, gris) se vendent bien.
4. **Comparaison avec les annonces proches** :
  - **eBay** : Les vestes neuves en laine Zara (taille S) oscillent entre **75 € et 165 €**.
  - **Vinted** : Les vestes en très bon état ou neuves (taille S) se situent entre **20 € et 60 €**.
  - **Leboncoin** : Les vestes similaires sont proposées entre **10 € et 35 €**, mais souvent en occasion.

**Un prix entre 60 € et 80 € est réaliste pour une vente équilibrée (ni trop longue, ni sous-évaluée).**

---

### **Annonces les plus proches (référence) :**
1. **eBay** :
  - "MANTEAU ZARA NEUF 100% LAINE AVEC CEINTURE [...] BLEU" → **165,34 €** (neuf, taille non précisée)
  - "ZARA NAVY BLUE DOUBLE-BREASTED 100% WOOL KNIT JACKET SIZE M" → **110,22 €** (neuf, taille M mais proche)
  - "VESTE EN TRICOT ZARA BLEU MARINE [...] 100% LAINE TAILLE M" → **75,78 €** (neuf, taille M)
2. **Vinted** :
  - "VESTE EN TRICOT ZARA BLEU MARINE [...] 100% LAINE TAILLE M" → **75,78 €** (neuf, taille M)
  - "Veste Zara" (neuf avec étiquette, taille S) → **20 €**
  - "Est en laine Zara" (neuf sans étiquette, taille S) → **35 €**
  - "Manteau Zara baby blue" (très bon état, taille L) → **27 €** (couleur proche)
3. **Leboncoin** :
  - "VESTE BLEU MARINE T36 FEMME" → **13 €** (occasion, taille S)
  - "MANTEAU CABAN LAINE BLEU MARINE" → **10 €** (occasion)

---

### **Recommandations pour la vente :**
- **Plateforme idéale** : **Vinted** (meilleur compromis prix/visibilité) ou **eBay** (si la veste est rare).
- **Photos** : Mettre en avant l'étiquette "neuf", la composition (100% laine) et les détails (boutons, coupe).
- **Description** : Préciser "neuf avec étiquette", la taille exacte, et les mots-clés comme "veste hiver femme", "laine chaude", "style élégant".
- **Prix de départ** : **75 €** (négociable à 60-65 € si besoin).

Si la veste a des détails uniques (doubleure, boutons originaux), le prix peut être ajusté à la hausse.

→ Résultat sauvegardé dans estimation_mistral.txt
PS C:\Users\rayan\OneDrive\Bureau\Projet_encheres >
```

```
EXPLORER    PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PROJET_ENCHERES 2
ebay.csv
estimation_mistral.txt
leboncoin.csv
mistral.py
produit.json
resultats.json
scraper.py
vinted.csv

OUTLINE
TIMELINE

Ln 1, Col 1  Spaces: 4  UTF-8  CRLF  Plain Text

- "VESTE EN TRICOT ZARA BLEU MARINE [...] 100% LAINE TAILLE M" → **75,78 €** (neuf, taille M)

2. **Vinted** :
- "Veste Zara" (neuf avec étiquette, taille S) → **20 €**
- "Est en laine Zara" (neuf sans étiquette, taille S) → **35 €**
- "Manteau Zara baby blue" (très bon état, taille L) → **27 €** (couleur proche)

3. **Leboncoin** :
- "VESTE BLEU MARINE T36 FEMME" → **13 €** (occasion, taille S)
- "MANTEAU CABAN LAINE BLEU MARINE" → **10 €** (occasion)

---

### **Recommandations pour la vente :**
- **Plateforme idéale** : **Vinted** (meilleur compromis prix/visibilité) ou **eBay** (si la veste est rare).
- **Photos** : Mettre en avant l'étiquette "neuf", la composition (100% laine) et les détails (boutons, coupe).
- **Description** : Préciser "neuf avec étiquette", la taille exacte, et les mots-clés comme "veste hiver femme", "laine chaude", "style élégant".
- **Prix de départ** : **75 €** (négociable à 60-65 € si besoin).

Si la veste a des détails uniques (doubleure, boutons originaux), le prix peut être ajusté à la hausse.

→ Résultat sauvegardé dans estimation_mistral.txt
PS C:\Users\rayan\OneDrive\Bureau\Projet_encheres >
```

8. Conclusion

Le développement de ce système a permis de mettre en place une solution complète, automatisée et fiable pour l'analyse de produits à partir d'annonces en ligne. Grâce à l'intégration cohérente entre le script Python, le workflow n8n, les outils de scraping comme Jina AI et l'analyse IA via Mistral, le pipeline fonctionne désormais de manière fluide, depuis la collecte des données jusqu'à la génération d'une estimation finale. Les tests réalisés ont confirmé la pertinence des résultats obtenus, ainsi que la capacité du système à s'adapter à différentes plateformes et à des données hétérogènes.

Ce projet a également été l'occasion de surmonter plusieurs défis techniques, notamment liés à la communication entre les modules, à la qualité des données collectées et aux limitations des outils utilisés. Les solutions apportées ont renforcé la robustesse du système et ont permis d'améliorer la collaboration entre les membres de l'équipe. Le résultat final est un outil capable de produire des estimations cohérentes, argumentées et basées sur des données réelles du marché.

Au-delà de son fonctionnement actuel, ce travail ouvre la voie à de nombreuses perspectives d'amélioration, telles que l'ajout de nouvelles plateformes, l'optimisation du scraping, l'enrichissement des analyses IA ou encore la mise en place d'une interface utilisateur. Le projet constitue ainsi une base solide pour des évolutions futures et démontre l'intérêt d'un pipeline automatisé combinant scraping, traitement de données et intelligence artificielle.

9. Perspectives

Le projet a permis de développer un pipeline automatisé complet pour la collecte et l'analyse de données issues de plateformes de vente en ligne. Plusieurs perspectives d'évolution peuvent être envisagées :

- **Ajout de nouvelles plateformes** : étendre le scraping à d'autres sites d'enchères ou de ventes pour enrichir les données et améliorer la pertinence des estimations de prix.
- **Optimisation du scraping** : intégrer des solutions plus robustes pour gérer les changements structurels des sites, réduire le risque d'échec et accélérer la collecte de données.
- **Enrichissement de l'analyse IA** : utiliser des modèles plus avancés pour la prédiction des prix, la détection des tendances du marché ou l'évaluation de la rareté et de la valeur des produits.
- **Interface utilisateur** : développer une interface graphique pour permettre à l'utilisateur de saisir ses produits, consulter les annonces similaires et obtenir l'estimation de manière intuitive.
- **Automatisation et fiabilité accrues** : limiter les interventions manuelles liées aux changements d'URL (ngrok) ou aux formats de données, et garantir la reproductibilité complète du pipeline.
- **Analyse prédictive avancée** : intégrer des outils d'analyse statistique et de machine learning pour identifier les facteurs influençant le prix final et proposer des recommandations personnalisées pour la vente.

Ces perspectives ouvrent la voie à un projet plus complet, capable de fournir une solution commerciale et technique robuste pour l'estimation automatisée de prix et l'analyse de produits sur différentes plateformes.

10. Bibliographie

- Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media.
- Raschka, S., & Mirjalili, V. (2019). *Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2*. Packt Publishing.
- Chollet, F. (2018). *Deep Learning with Python*. Manning Publications.
- McKinney, W. (2018). *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. O'Reilly Media.
- Russell, S., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach* (4th edition). Pearson.

11.Webographie

- eBay Developers Program – <https://developer.ebay.com>
- Vinted API Documentation – <https://www.vinted.fr/developers>
- Leboncoin Developers – <https://developer.leboncoin.fr>
- n8n Documentation – <https://docs.n8n.io>
- Jina AI Documentation – <https://docs.jina.ai>
- Mistral AI API – <https://www.mistral.ai>
- Docker Documentation – <https://docs.docker.com>
- BeautifulSoup Documentation – <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
- Selenium Documentation – <https://www.selenium.dev/documentation/>

12. Annexes

A. Diagramme d'architecture du système

- Illustration du pipeline complet :
 - Script Python principal (saisie des caractéristiques, génération de JSON)
 - Workflow n8n (scraping, nettoyage et consolidation des annonces)
 - Module d'analyse IA (estimation des prix via Mistral)
- Permet de visualiser le flux de données entre les différents composants et leur interaction.

B. Exemples de fichiers CSV générés

- **shopgoodwill_<motclé>.csv** : annonces récupérées avec titre, prix, URL et plateforme.
- **CSV global** regroupant toutes les plateformes pour une même requête.
- Utile pour tester la cohérence des données et la reproductibilité de l'analyse.

C. Exemples d'annonces récupérées

- **Annonces les plus proches (référence)** :
 - eBay, Vinted, Leboncoin (avec titres, prix et état)
- Permet de vérifier que le système trouve des annonces pertinentes et proches du produit étudié.

D. Capture d'écran du système en fonctionnement

- Affichage des annonces, du traitement et de l'estimation finale fournie par Mistral.
- Permet d'illustrer la fluidité et la cohérence de l'interface utilisateur du pipeline.

E. Extraits de scripts Python et workflow n8n

- Extraits commentés pour :
 - La collecte des caractéristiques
 - La génération des fichiers JSON et CSV
 - L'interrogation de l'API Mistral
- Workflow n8n : capture de l'enchaînement des nœuds de scraping et de nettoyage.

F. Exemples de tests et résultats

- Test avec mot-clé "looney" : récupération de 120 annonces réparties sur 3 pages.
- Vérification des prix, titres et correspondances entre plateformes.
- Ces annexes permettent d'évaluer la pertinence du système et la fiabilité des estimations.