

# Programme d'Ingénierie en Intelligence Artificielle – OC IA P7 -

## Analyse de sentiments grâce au Deep Learning

### Préambule

Ce support de présentation est spécifiquement **destiné à une audience non technique**, comme indiqué dans l'énoncé du projet. Il s'adresse principalement à des responsables marketing, décideurs stratégiques, et autres parties prenantes intéressées par les résultats concrets et les applications pratiques du projet.

L'objectif de cette présentation est de vulgariser les aspects complexes de manière claire et accessible. Elle mettra l'accent sur :

- Une vue d'ensemble des données (tweets) et des **traitements réalisés**,
- Les **résultats des modèles** développés pour prédire les sentiments (positifs ou négatifs),
- La **mise en production** d'un prototype opérationnel pouvant être utilisé dans un contexte réel.

Bien que simplifiée pour être compréhensible par tous, cette présentation est appuyée par une démarche rigoureuse. Les détails techniques approfondis (par exemple, le fonctionnement des modèles ou le suivi des performances) sont disponibles dans les documents techniques associés.

Nous vous invitons à poser des questions tout au long de la présentation pour clarifier tout concept ou résultat.

# Introduction

## Objectif du projet

Développer un outil capable de comprendre automatiquement les sentiments (positifs ou négatifs) exprimés dans des tweets, afin d'aider Air Paradis à détecter rapidement des situations de bad buzz.

## Résultats clés

- **Modèle utilisé** : Une méthode rapide et efficace basée sur **FastText**, permettant d'obtenir une précision de 82 %.
- **Outil pratique** : Une application connectée via une **API** pour analyser les tweets en temps réel, déployée sur **Microsoft Azure**.
- **Automatisation** : Un système fiable de mise à jour continue grâce à des outils comme **GitHub Actions** et **Docker**.
- **Suivi** : Un tableau de bord pour surveiller les performances et recevoir des alertes en cas d'erreur.

## Étapes principales

- **Préparation** des données en nettoyant et organisant les tweets.
- **Comparaison de plusieurs solutions** pour trouver la méthode la plus efficace.
- **Création et déploiement d'un outil opérationnel** pour analyser les tweets en temps réel.

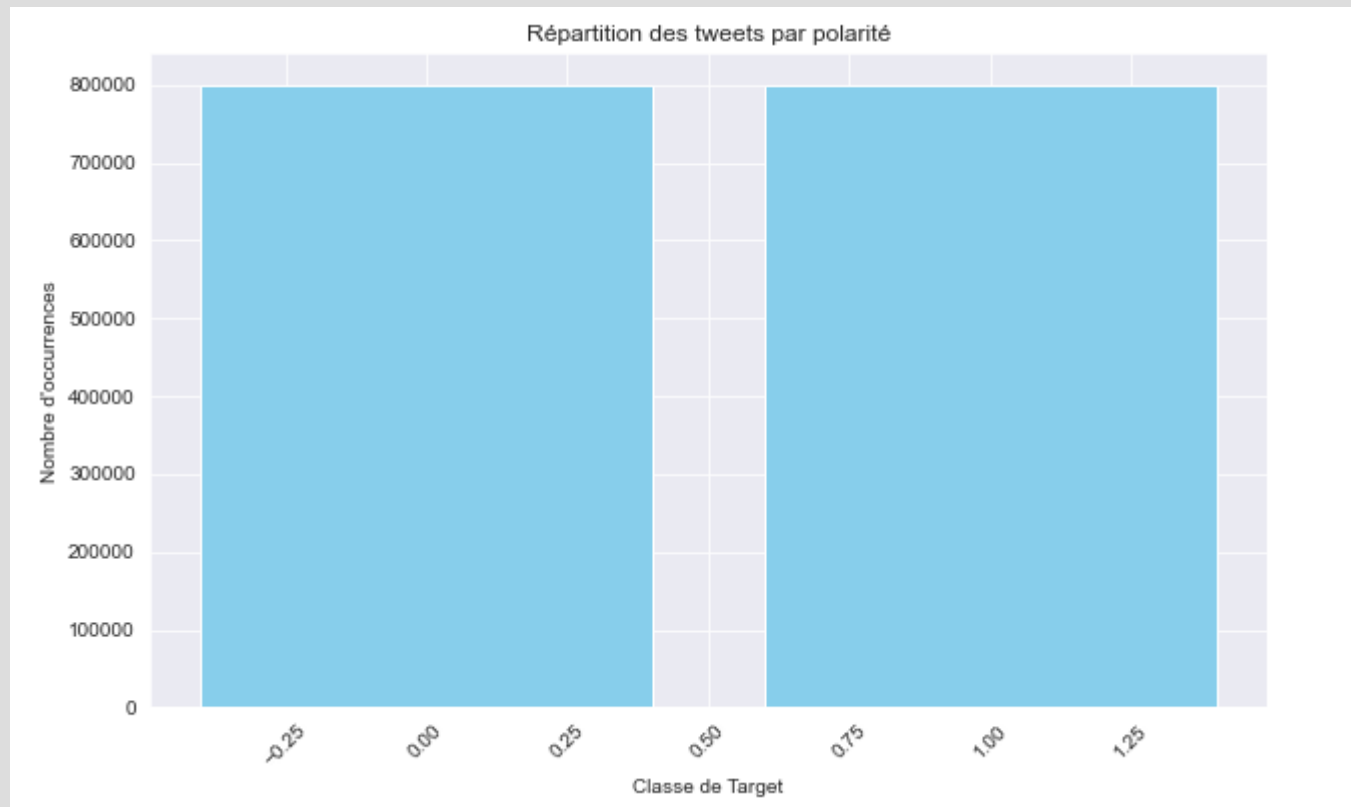
# Sommaire

- 1) EDA
- 2) Élaboration des modèles
- 3) Comparaison des résultats
- 4) Principes du MLOps
- 5) Mise en production
- 6) Conclusion et perspectives

# 1) EDA (Exploratory Data Analysis)

L'EDA consiste à explorer, visualiser et comprendre les données avant de les utiliser pour la modélisation ou l'analyse.

Nous avons à notre disposition un jeu de données Open Source, celui ci contient 1 600 000 tweets labellisés avec un sentiment (positif ou négatif).



# 1) EDA (Exploratory Data Analysis)

La **distribution** des Tweets positifs et négatifs est **strictement équilibrée**, ce qui constitue un **scénario optimal pour l'entraînement de modèles de machine learning**. Cela réduit les risques de biais induits par une sur-représentation ou sous-représentation.

## Prétraitement des données : Lemmatisation des tweets

La lemmatisation consiste à ramener les mots à leur forme de base ("lemme") pour uniformiser le texte et réduire les variations inutiles. Cela simplifie l'analyse et améliore l'efficacité des modèles d'apprentissage.

## Exemple de tweet avant/après lemmatisation :

### Avant :

"The flights were delayed, and I am waiting at the airport for hours. Hoping things get better soon."

### Après :

"the flight be delay and I be wait at the airport for hour hope thing get good soon"

## Pourquoi est-ce important ?

Réduit la taille du vocabulaire analysé, rendant les calculs plus efficaces.

Améliore la capacité des modèles à détecter les sentiments en supprimant les variations inutiles des mots.

## 2) Élaboration des modèles

### Approches testées

Pour prédire les sentiments des tweets, nous avons exploré trois approches différentes :

#### **Modèle simple :**

Une régression logistique, rapide à entraîner mais limitée dans sa capacité à comprendre les contextes complexes des phrases.

#### **Modèle avancé avec FastText :**

Un réseau de neurones utilisant des représentations textuelles avancées (embeddings FastText), permettant de capturer des relations sémantiques entre les mots.

#### **Modèle BERT :**

Un modèle pré-entraîné sur de grands volumes de données, capable de comprendre le contexte précis des phrases, mais exigeant en ressources.

### Qu'est-ce qu'un embedding ?

Les embeddings transforment les mots en nombres, permettant aux modèles de machine learning de "comprendre" leur signification.

#### **Exemple simplifié :**

Le mot "avion" → [0.21, 0.43, -0.12]

Les mots proches, comme "aéroport", auront des valeurs similaires, ce qui aide à capturer les relations entre les mots.

### Étapes principales pour élaborer un modèle

**Préparation des données :** Nettoyage et transformation des tweets en embeddings.

**Création du modèle :** Entraînement avec les tweets pour associer un sentiment (positif ou négatif).

**Évaluation :** Test des performances pour choisir le modèle le plus adapté.

## 2) Élaboration des modèles

### MLFlow - Visualisation des expérimentations : capture d'écran de l'interface MLFlow montrant les différentes expérimentations

The screenshot displays the MLFlow web interface at the URL `127.0.0.1:5000/#/compare-experiments/s?experiments=%5B%22711971181331744008%2C%22793213831385266638%2C%22835664034872152055%2C%22974948647011922974%22%5D&searchFilter=&...`. The interface includes a top navigation bar with the MLFlow logo (2.17.0) and tabs for Experiments and Models. A sidebar on the left lists four experiments: BERT\_Embedding\_Experiment, Logistic\_Regression\_TFIDF\_Experiment, FastText\_Embedding\_Experiment, and GloVe\_Embedding\_Experiment. The main content area is titled "Displaying Runs from 4 Experiments" and features tabs for Runs, Evaluation, Experimental, and Traces. A search bar contains the query `metrics.rmse < 1 and params.model = "tree"`. Below the search bar, a table lists the runs with columns for Run Name, Created, Dataset, Duration, Source, and Models. The table shows eight runs, with the first four having a duration of 24.7s, 8.8s, 9.1min, and 5.3min respectively. The last four runs have a duration of 45ms, 49ms, 7.2min, and 5.6min. A "Show more columns (19 total)" button is visible on the right side of the table.

**Experiments**

- BERT\_Embedding\_Experiment
- Logistic\_Regression\_TFIDF\_Experiment
- FastText\_Embedding\_Experiment
- GloVe\_Embedding\_Experiment

**Displaying Runs from 4 Experiments**

Runs Evaluation Experimental Traces Experimental

Search Experiments

metrics.rmse < 1 and params.model = "tree"

Time created State: Active Datasets Sort: Created Columns Group by

Run Name	Created	Dataset	Duration	Source	Models
awesome-grouse-454	1 month ago	-	24.7s	C:\Users...	keras
trusting-worm-774	1 month ago	-	8.8s	C:\Users...	sklearn
FastText_Model_Evaluation	1 month ago	-	9.1min	C:\Users...	keras
GloVe_Model_Evaluation	1 month ago	-	5.3min	C:\Users...	keras
GloVe_Model_Evaluation	1 month ago	-	45ms	C:\Users...	-
GloVe_Model_Evaluation	1 month ago	-	49ms	C:\Users...	-
FastText_Model_Evaluation	1 month ago	-	7.2min	C:\Users...	keras
GloVe_Model_Evaluation	1 month ago	-	5.6min	C:\Users...	keras

Show more columns (19 total)

### 3) Comparaison des résultats

#### Les critères d'évaluation :

**Précision (Accuracy)** : Combien de tweets ont été correctement classés.

**Fiabilité (F1-score)** : Évalue la cohérence des résultats, même lorsque les données sont déséquilibrées.

Imaginez que chaque modèle est comme un traducteur (utilise les embeddings pour transformer les mots en données compréhensibles pour la machine) :

**Régression logistique** est un traducteur qui ne capte que des mots individuels.

**FastText + LSTM** interprète mieux les phrases complètes et leur signification..

**BERT**, bien que très précis, est plus lent car il analyse chaque mot en détail.

#### Comparatif des performances :

Approche	Accuracy	F1-score
Régression logistique	0.770	0.770
GloVe + LSTM	0.830	0.820
FastText + LSTM	0.820	0.830
BERT (bert-base-uncased)	0.800	0.790

#### Choix final : FastText + LSTM

##### Pourquoi ?

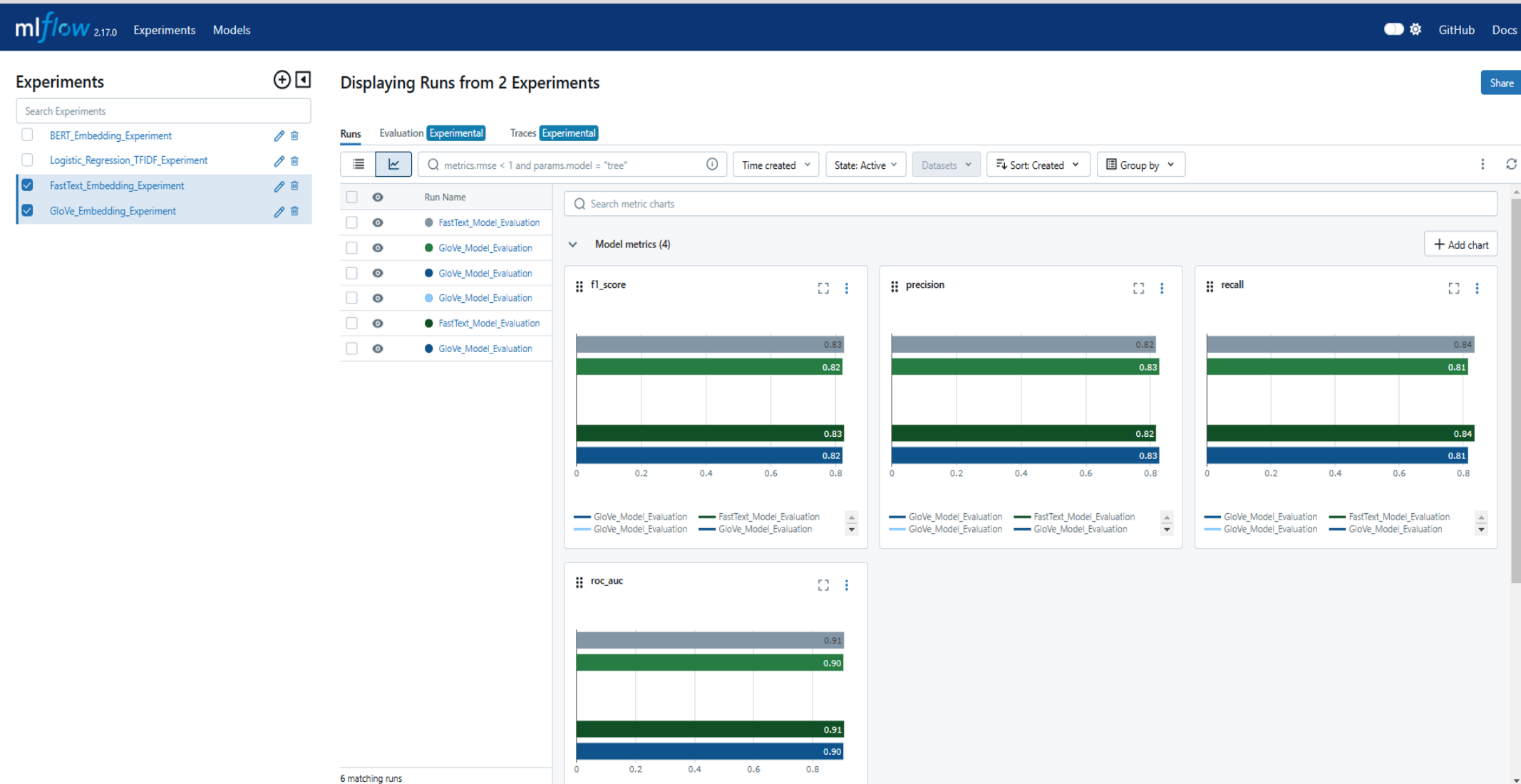
Performance proche de la meilleure méthode.

Moins gourmand en ressources, idéal pour un déploiement rapide .



### 3) Comparaison des résultats

capture de l'interface MLFlow montrant la **comparaison des modèles** en termes de précision et F1-score



## 4) Principes du MLOps

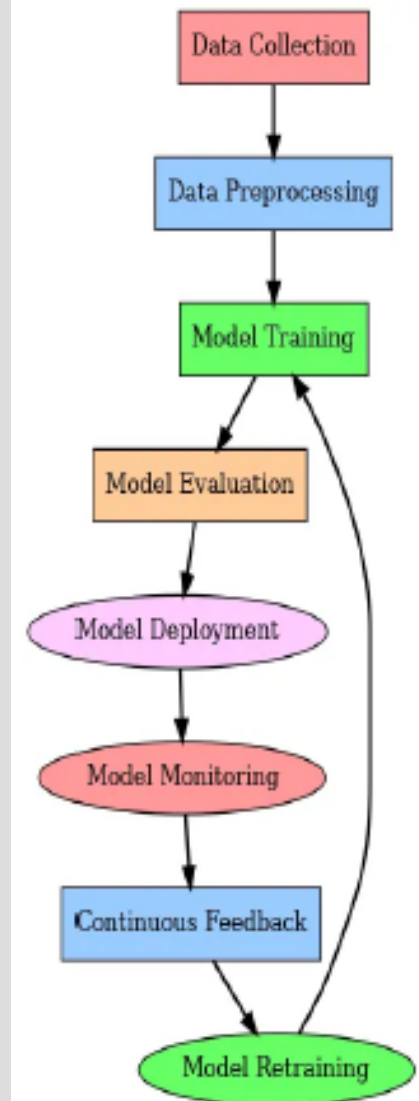
### Qu'est-ce que le MLOps ?

Le MLOps (**Machine Learning Operations**) est une méthode qui combine les principes de la gestion des projets techniques (DevOps) avec les besoins spécifiques des projets d'apprentissage automatique. Cela permet de rendre le déploiement et la gestion des modèles d'intelligence artificielle plus fiables, rapides et automatisés.

Le diagramme illustre le **cycle de vie d'un modèle d'IA** :

- Les données sont **collectées et stockées** (via un fichiers CSV open source),
- puis **nettoyées et prétraitées** à l'aide de **Python** pour la préparation des données,
- utilisées pour **entraîner** un modèle, avec **MLFlow** pour le suivi des expérimentations,
- **validées lors de l'évaluation** grâce aux métriques de performances (précision et F1-score),
- **déployées** en production sur **Microsoft Azure App Service** avec **Docker** et un pipeline **CI/CD GitHub Actions**,
- **surveillées via Application Insights** pour détecter des erreurs ou anomalies,
- et régulièrement **améliorées et réentraînées** grâce aux nouvelles données et aux retours des utilisateurs.

Ce processus garantit que le modèle reste performant dans le temps.



## 5) Mise en production

### Glossaire (Mise en production)

- **API** : Interface permettant de connecter le modèle à une application ou un service pour envoyer des tweets et obtenir des prédictions.
- **FastAPI** : Outil rapide et flexible pour créer des API performantes.
- **CI/CD** : Processus d'automatisation pour tester et déployer les mises à jour du modèle ou du code sans intervention manuelle.
- **Docker** : Technologie qui permet de "packager" le modèle pour qu'il fonctionne sur tout type de plateforme.
- **Azure App Service** : Service cloud utilisé pour héberger l'API en ligne et la rendre accessible aux utilisateurs.
- **Application Insights** : Outil de surveillance en temps réel pour suivre les performances de l'API et détecter les erreurs.

## 5) Mise en production

Pour mettre le modèle en production, nous avons réalisé **plusieurs étapes clés** :

### - Déploiement de l'API :

Le modèle prédictif a été intégré dans une API développée avec **FastAPI** et déployée sur **Microsoft Azure App Service**. Cette API permet de recevoir un tweet en entrée et de retourner un sentiment (positif ou négatif).

### - Automatisation des déploiements :

Un pipeline **CI/CD (Intégration Continue / Déploiement Continu)** a été mis en place avec **GitHub Actions**. Cela garantit que toute mise à jour du modèle ou du code est automatiquement testée et déployée.

**Docker** a été utilisé pour encapsuler le modèle dans un conteneur, rendant le déploiement fiable et reproductible, peu importe l'environnement.

### - Surveillance des performances :

Nous utilisons **Azure Application Insights** pour suivre les performances en temps réel. Cela inclut la détection des prédictions incorrectes et le déclenchement d'alertes, comme l'envoi d'un email en cas de problème.

### - Test de l'API :

Une interface interactive via **Swagger** permet de visualiser les endpoints de l'API et de tester son fonctionnement directement.

## 5) Mise en production

Capture d'écran de l'historique des commits sur GitHub ([https://github.com/preudh/OC\\_IA\\_P7\\_analyse\\_sentiments\\_deep\\_Learning.git](https://github.com/preudh/OC_IA_P7_analyse_sentiments_deep_Learning.git)).

github.com/preudh/OC\_IA\_P7\_analyse\_sentiments\_deep\_Learning/actions

preudh / OC\_IA\_P7\_analyse\_sentiments\_deep\_Learning

Code Issues Pull requests **Actions** Projects Security Insights Settings

**Actions** New workflow

All workflows

CI/CD Pipeline with Build and Deploy Disabled

Management

- Caches
- Deployments ↗
- Attestations ↗
- Runners

All workflows

Showing runs from all workflows

Filter workflow runs

49 workflow runs

Event	Status	Branch	Actor
✓ docker compose ci	CI/CD Pipeline with Build and Deploy #49: Commit <a href="#">448ba0a</a> pushed by preudh	master	last week 4m 16s
✗ APPINSIGHTS_INSTRUMENTATIONKEY docker compose	CI/CD Pipeline with Build and Deploy #48: Commit <a href="#">d2f763b</a> pushed by preudh	master	last week 37s
✗ Add Azure Application Insights integration file deploy	CI/CD Pipeline with Build and Deploy #47: Commit <a href="#">d5fd346</a> pushed by preudh	master	last week 1m 33s
✗ Add Azure Application Insights integration	CI/CD Pipeline with Build and Deploy #46: Commit <a href="#">f5a1ee9</a> pushed by preudh	master	last week 1m 29s
✓ Merge remote-tracking branch 'origin/master'	CI/CD Pipeline with Build and Deploy #45: Commit <a href="#">c6495e4</a> pushed by preudh	master	last week 6m 5s
✓ Update deploy.yml	CI/CD Pipeline with Build and Deploy #44: Commit <a href="#">f9b203b</a> pushed by preudh	master	last week 4m 7s
✓ Update deploy.yml	CI/CD Pipeline with Build and Deploy #43: Commit <a href="#">85d71a5</a> pushed by preudh	master	2 weeks ago 3m 53s
✓ deploy.yml pour inclure la construction et le push corrects	CI/CD Pipeline with Build and Deploy #42: Commit <a href="#">e0f0581</a> pushed by preudh	master	2 weeks ago 3m 11s
✓ compose ci main	CI/CD Pipeline with Build and Deploy #41: Commit <a href="#">cc5dd85</a> pushed by preudh	master	2 weeks ago 2m 44s
✓ Dockerfile	CI/CD Pipeline with Build and Deploy #40: Commit <a href="#">450278c</a> pushed by preudh	master	2 weeks ago 2m 20s
✓ Dockerfile pour que l'application écoute sur le port 80 au lieu de 8000.	CI/CD Pipeline with Build and Deploy #39: Commit <a href="#">9312937</a> pushed by preudh	master	2 weeks ago 2m 8s

## 5) Mise en production

Capture de l'exécution des **tests unitaires** dans **GitHub Actions** et des **étapes du pipeline CI/CD**, assurant la qualité et la fiabilité avant déploiement.

The screenshot displays the GitHub Actions interface for a workflow named 'CI/CD Pipeline with Build and Deploy'. The specific run is 'docker compose ci #49', which has a green checkmark indicating success. The interface includes a top navigation bar with links to Code, Issues, Pull requests, Actions (selected), Projects, Security, Insights, and Settings. A search bar is located in the top right corner.

On the left sidebar, the 'Summary' tab is active, showing a list of jobs: 'test' and 'build-and-deploy', both marked with green checkmarks. Below this, there are links for 'Run details', 'Usage', and 'Workflow file'.

The main content area provides a summary of the run:

- Triggered via push last week
- Status: Success
- Total duration: 4m 16s
- Artifacts: -

Below the summary, the workflow steps are visualized in a sequence diagram:

```
graph LR; test[✓ test 1m 25s] --> build-and-deploy[✓ build-and-deploy 2m 32s]
```

The diagram shows two steps: 'test' (duration 1m 25s) and 'build-and-deploy' (duration 2m 32s), both marked with green checkmarks. The workflow is triggered 'on: push'.

# 5) Mise en production

## Vue d'ensemble des ressources sur Microsoft Azure

Capture des services utilisés pour le déploiement et le suivi du modèle en production.

Microsoft Azure

Rechercher dans les ressources, services et documents (G+/)

Copilot

preudhomme.patrice@...  
RÉPERTOIRE PAR DÉFAUT (PREU...

### Services Azure

[Créer une ressource](#) [Surveillance](#) [Alertes](#) [Aide et support](#) [GitHub](#) [Toutes les ressources](#) [App Services](#) [Abonnements](#) [Microsoft Entra ID](#) [Autres services](#)

### Ressources

Récent Favori

Nom	Type	Dernier affichage
p7-myapp-deploy	Application Insights	il y a 2 jours
Application Insights Smart Detection	Groupe d'actions	il y a 2 jours
Alerte sur les prédictions incorrectes des tweets	Règle d'alerte de recherche dans les journaux	il y a 2 jours
p7-myapp-deploy	App Service	il y a 6 jours
myappregistry1	Registre de conteneurs	il y a une semaine
p7-myapp-deploy/fchea8befabmcxhk	App Service (emplacement)	il y a une semaine
ASP-MyAppResources-8b78	Plan App Service	il y a une semaine
Abonnement Azure 1	Abonnement	il y a une semaine
MyAppResources	Groupe de ressources	il y a une semaine

[Tout afficher](#)

### Naviguer

[Abonnements](#) [Groupes de ressources](#) [Toutes les ressources](#) [Tableau de bord](#)

### Outils

**Microsoft Learn**  
Se former sur Azure avec les formations en ligne gratuites proposées par Microsoft

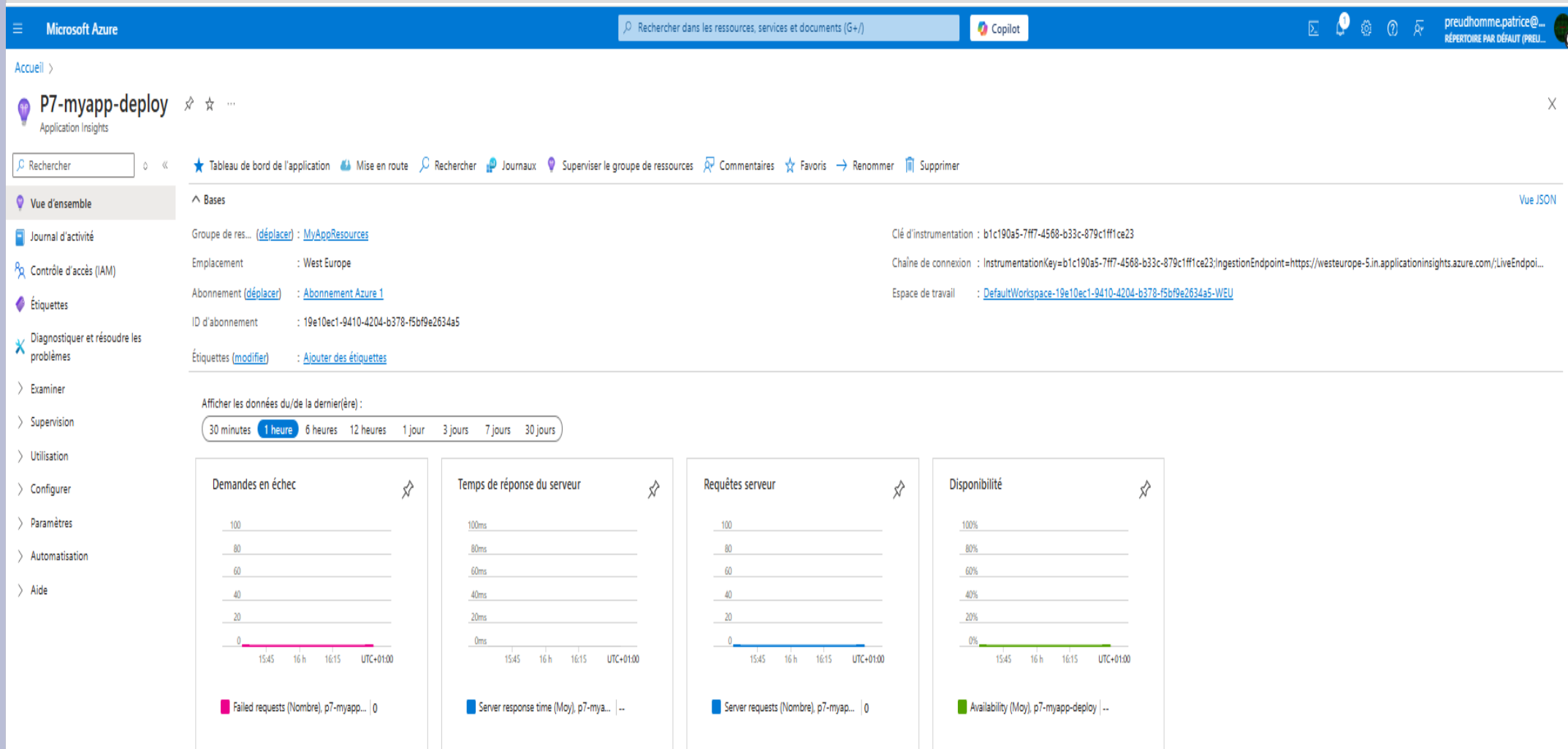
**Azure Monitor**  
Assurer le monitoring de vos applications et infrastructure

**Microsoft Defender pour le cloud**  
Sécuriser vos applications et votre infrastructure

**Gestion des coûts**  
Analyser et optimiser gratuitement vos dépenses cloud

## 5) Mise en production

Capture d'écran montrant les logs ou les graphiques de **suivi des performances via Azure Application Insights**.





# 5) Mise en production

Capture de l'alerte configurée (en cas de prédictions incorrectes).

Microsoft Azure

Rechercher dans les ressources, services et documents (G+)

Copilot

preudhomme.patrice@...

RÉPERTOIRE PAR DÉFAUT (PREU...

Accueil > Alerte sur les prédictions incorrectes des tweets

Alerte sur les prédictions incorrectes des tweets | Historique

☆ ...

Règle d'alerte de recherche dans les journaux

Rechercher

Rechercher

Intervalle de temps : 7 derniers jours

Vue d'ensemble

Journal d'activité

Contrôle d'accès (IAM)

Étiquettes

Diagnostiquer et résoudre les problèmes

Historique

Paramètres

Configuration des règles d'alerte

Automatisation

Aide

0 - Critique

1 - Erreur

2 - Avertissement

3 - Information

4 - Commentaires

Aucune alerte

Chronologie

22 nov. 12:00 23 nov. 12:00 24 nov. 12:00 25 nov. 12:00 26 nov. 12:00 27 nov. 12:00 28 nov. 12:00

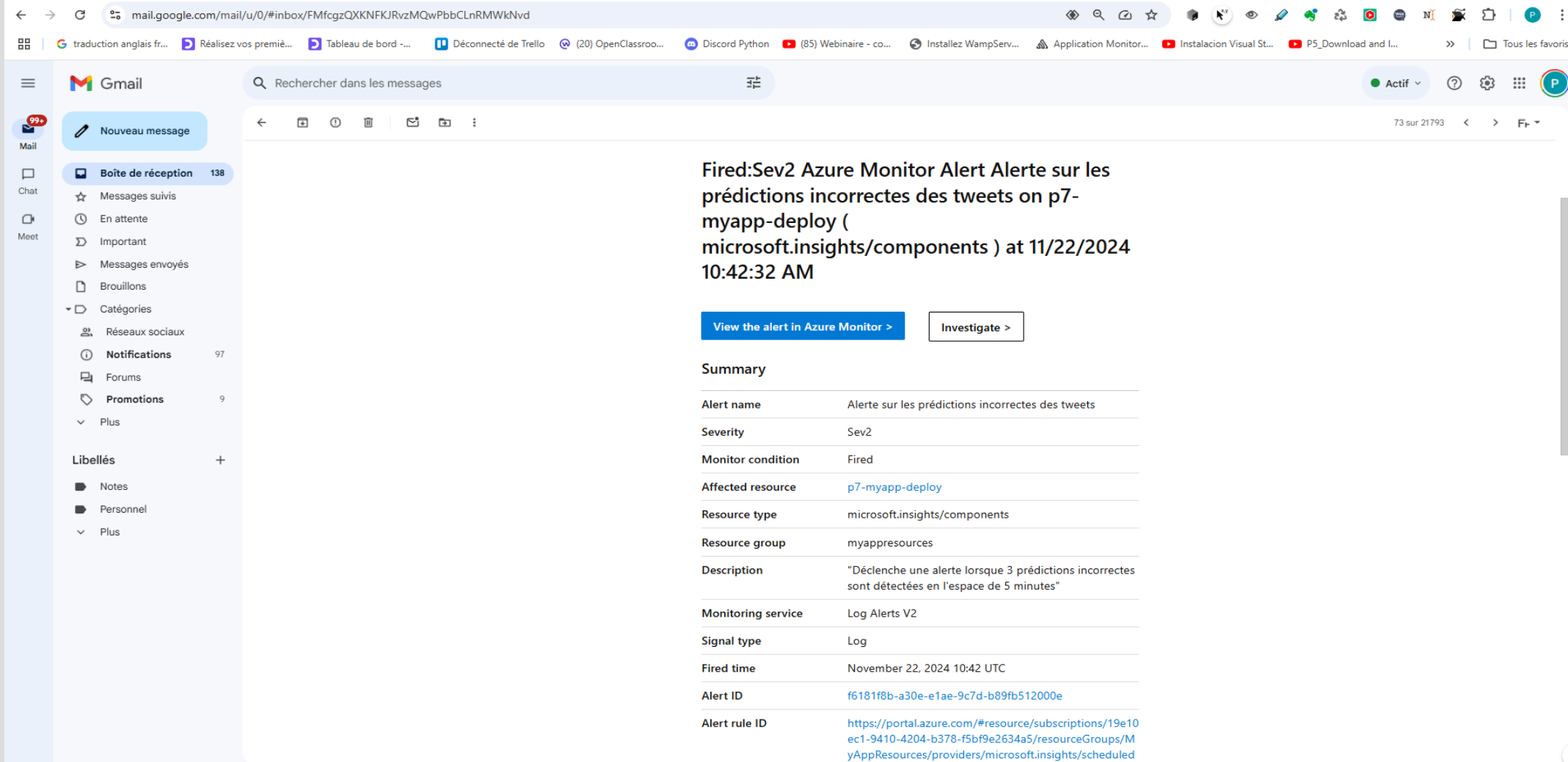
2

Heure de déclenchement	Condition d'alerte	Ressource affectée	Réponse de l'utilisateur	Valeur (en cas de déclenchement de l'al...	Dimensions
22/11/2024 11:42	⚠ Déclenché	📍 p7-myapp-deploy	📄 Nouveau	1	Aucun
22/11/2024 11:37	⚠ Déclenché	📍 p7-myapp-deploy	📄 Nouveau	1	Aucun

## 5) Mise en production

### Suivi des erreurs et alertes :

L'outil Azure Application Insights surveille les prédictions en temps réel. **En cas de 3 prédictions incorrectes en moins de 5 minutes, une alerte est déclenchée et un email est envoyé.**



The screenshot shows a Gmail interface with a search bar at the top. The left sidebar contains navigation options: Mail (99+), Chat, and Meet. The main content area displays an email titled "Fired:Sev2 Azure Monitor Alert Alerte sur les prédictions incorrectes des tweets on p7-myapp-deploy ( microsoft.insights/components ) at 11/22/2024 10:42:32 AM". Below the title are two buttons: "View the alert in Azure Monitor >" and "Investigate >". The email body contains a "Summary" section with the following details:

Field	Value
Alert name	Alerte sur les prédictions incorrectes des tweets
Severity	Sev2
Monitor condition	Fired
Affected resource	<a href="#">p7-myapp-deploy</a>
Resource type	microsoft.insights/components
Resource group	myappresources
Description	"Déclenche une alerte lorsque 3 prédictions incorrectes sont détectées en l'espace de 5 minutes"
Monitoring service	Log Alerts V2
Signal type	Log
Fired time	November 22, 2024 10:42 UTC
Alert ID	<a href="#">f6181f8b-a30e-e1ae-9c7d-b89fb512000e</a>
Alert rule ID	<a href="https://portal.azure.com/#resource/subscriptions/19e10ec1-9410-4204-b378-f5bf9e2634a5/resourceGroups/MyAppResources/providers/microsoft.insights/scheduled">https://portal.azure.com/#resource/subscriptions/19e10ec1-9410-4204-b378-f5bf9e2634a5/resourceGroups/MyAppResources/providers/microsoft.insights/scheduled</a>

## 5) Mise en production

### Accès à l'API via Swagger :

L'API a été développée avec **FastAPI** et déployée sur **Microsoft Azure App Service**. L'interface Swagger permet de visualiser et de tester les différents endpoints de l'API, comme illustré ci-contre avec le domaine <https://p7-myapp-deploy-fchea8befabmcxhk.westeurope-01.azurewebsites.net/docs>

Exemple : En envoyant un tweet via Swagger, l'API renvoie une prédiction indiquant si le tweet est positif ou négatif .

The screenshot shows a web browser window displaying the Swagger UI for a FastAPI application. The browser's address bar shows the URL: `p7-myapp-deploy-fchea8befabmcxhk.westeurope-01.azurewebsites.net/docs`. The Swagger UI header includes the text "FastAPI" with version indicators "0.1.0" and "OAS 3.1", and a link to the OpenAPI JSON file: `/openapi.json`.

The "default" section lists two endpoints:

- POST /predict** Predict
- POST /feedback** Feedback

The "Schemas" section lists four data models:

- FeedbackInput** > Expand all object
- HTTPValidationError** > Expand all object
- TextInput** > Expand all object
- ValidationError** > Expand all object

## 6) Conclusion

Au cours de ce projet, nous avons **développé un modèle prédictif permettant d'identifier les sentiments exprimés dans des tweets. Plusieurs étapes clés ont permis de mener à bien cette réalisation.**

Nous avons d'abord **exploré différentes approches pour construire notre modèle**, en utilisant des techniques variées comme la régression logistique, FastText et BERT. Cette démarche comparative nous a permis de sélectionner le modèle **FastText**, offrant un excellent compromis entre précision, rapidité et consommation de ressources.

Pour la mise en production, nous avons intégré le modèle dans une **API déployée sur Microsoft Azure**. Un **pipeline CI/CD avec GitHub Actions et Docker** garantit la fiabilité et l'automatisation des mises à jour. Le suivi des performances en production est assuré grâce à **Azure Application Insights**, avec un **système d'alerte configuré pour détecter les prédictions incorrectes**.

### Perspectives d'évolution :

- **Renforcer la capacité du modèle à comprendre les subtilités des tweets** en explorant des technologies avancées telles que **GPT (Generative Pre-trained Transformer)**, un modèle de langage capable de comprendre des contextes complexes et de générer des prédictions extrêmement précises grâce à son entraînement sur de vastes bases de données textuelles.
- **Intégrer de nouvelles données** pour garantir l'adaptabilité aux changements des besoins stratégiques et affiner la pertinence des prédictions.
- **Simplifier davantage les processus de déploiement** en exploitant des services dédiés à l'intelligence artificielle tels que **Azure Machine Learning ou Azure OpenAI Service**, pour réduire les délais et les ressources nécessaires tout en bénéficiant d'infrastructures spécialisées.

Cette démarche pose les bases solides pour des développements futurs, tout en assurant une application robuste et évolutive.