

Hackathon : Analyse des tweets clients d'Engie et paramétrage d'Agents IA

Introduction

Dans ce hackathon, nous allons exploiter un fichier Excel contenant des tweets adressés au service client d'Engie afin de réaliser une analyse fine des problématiques rencontrées par les clients. L'objectif est d'extraire des indicateurs clés de performance (KPI), d'effectuer une analyse de sentiment et de paramétrer des agents IA sur Mistral ou Gemini pour automatiser la catégorisation des problèmes. Enfin, nous allons restituer ces analyses sous forme de tableau de bord interactif avec Power BI ou Streamlit.

Jour 1 - Objectifs de l'Hackathon

1. **Prétraitement des données avec Python et Pandas**
 - Charger et nettoyer les données
 - Convertir correctement les formats de date et d'identifiant
 - Extraire des informations utiles (heure de publication, longueur du texte, présence de mots-clés...)
2. **Calcul des KPI pertinents**
 - Nombre de tweets par jour/semaine/mois
 - Fréquence des mentions des comptes Engie
 - Détection des tweets contenant des mots-clés critiques ("délai", "panne", "urgence", "scandale", etc.)
3. **Analyse de sentiment des tweets**
 - Implémentation d'un modèle de sentiment analysis (à l'aide d'agent sur Mistral/Gemini)
 - Classification des tweets en trois catégories : Positif, Neutre, Négatif
 - Visualisation de la répartition des sentiments
4. **Paramétrage d'agents IA sur Mistral ou Gemini**
 - Entraîner un agent IA à détecter les types de réclamations

L'agent IA devra catégoriser les tweets selon les problématiques soulevées. Quelques catégories possibles :

- **Problèmes de facturation** : erreurs de montant, prélèvements injustifiés.
- **Pannes et urgences** : absence de gaz, d'électricité, problème d'eau chaude.
- **Service client injoignable** : absence de réponse, relances infructueuses.
- **Problèmes avec l'application** : bugs, indisponibilité du service.

- **Délai d'intervention** : retards dans la gestion des dossiers ou des réparations.
 - Calculer un score d'inconfort entre 0 et 100%
 - Générer des réponses automatiques basées sur les patterns identifiés (bonus)
 - 5. **Création d'un tableau de bord interactif avec Power BI ou Streamlit**
 - Afficher les principaux KPI sous forme de graphiques
 - Intégrer des filtres pour naviguer dans les données (par période, type de plainte...)
 - Restituer l'analyse de sentiment en temps réel
-

Consignes et pistes pour les étudiants

- **Nettoyage des données** : Attention aux valeurs manquantes et au format des identifiants numériques qui sont souvent stockés en notation scientifique.
 - **Approche NLP** : Utilisez des bibliothèques comme `nltk`, `spacy`, ou `transformers` pour traiter le texte.
 - **Exploration des données** : Pensez à générer des nuages de mots pour identifier les thèmes récurrents.
 - **Tableau de bord** : Utilisez Power BI si vous voulez une approche low-code ou Streamlit pour une solution Python interactive.
 - **Automatisation avec IA** : Testez différentes stratégies d'optimisation des prompts.
-

Livrables attendus

Les participants devront livrer un dépôt GitHub contenant :

- Un ou plusieurs scripts Python pour charger et nettoyer les données, extraire les KPI, analyser les sentiments des tweets et paramétrer les agents IA sur Mistral ou Gemini.
- Un fichier Power BI (.pbix) s'ils choisissent cette option pour le tableau de bord.
- Un README.md détaillant :
 - La méthodologie utilisée pour le traitement des tweets.
 - Les KPI retenus et leur signification.
 - L'approche employée pour l'analyse de sentiment.
 - Le processus de création des agents IA, incluant :
 - La logique utilisée pour détecter les types de réclamations.
 - Les prompts ou fine-tuning réalisés.
 - Des exemples d'interactions générées par l'agent.
 - Les choix technologiques pour la visualisation (Power BI ou Streamlit) et comment exécuter le projet.

L'ensemble du travail devra être documenté pour assurer sa bonne compréhension et permettre une éventuelle réutilisation ou amélioration.

Jour 2 - Consignes pour la Présentation et l'Analyse des Résultats

1. Présentation du travail réalisé (5 min)

- Expliquer brièvement la démarche adoptée : nettoyage des données, calcul des KPI, analyse de sentiment, paramétrage de l'agent IA.
- Présenter les outils utilisés et les choix techniques (bibliothèques, modèles NLP, approche de catégorisation des réclamations).
- Décrire le tableau de bord interactif et ses fonctionnalités clés.

2. Analyse des KPI et enseignements (10 min)

- Présenter les principaux indicateurs calculés (répartition des plaintes, fréquence des mentions, score d'inconfort...).
- Identifier les tendances majeures observées dans les données (pics d'activité, types de plaintes dominants...).
- Mettre en évidence les catégories les plus problématiques et les insights clés issue de l'analyse.

3. Recommandations et perspectives (5 min)

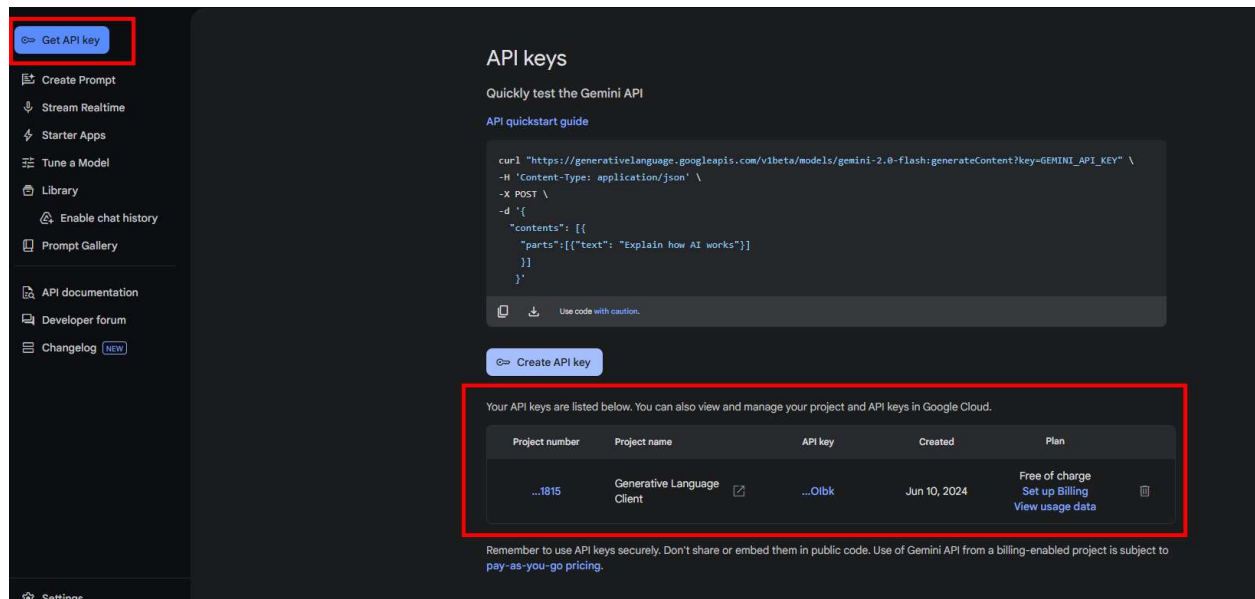
- Proposer des améliorations pour la gestion des réclamations en fonction des résultats obtenus.
- Suggérer des optimisations pour l'agent IA (meilleure détection des urgences, affinement des catégories...).
- Ouvrir sur des perspectives d'évolution : intégration des analyses en temps réel, amélioration du modèle NLP, automatisation de la réponse client.

Bon hackathon !

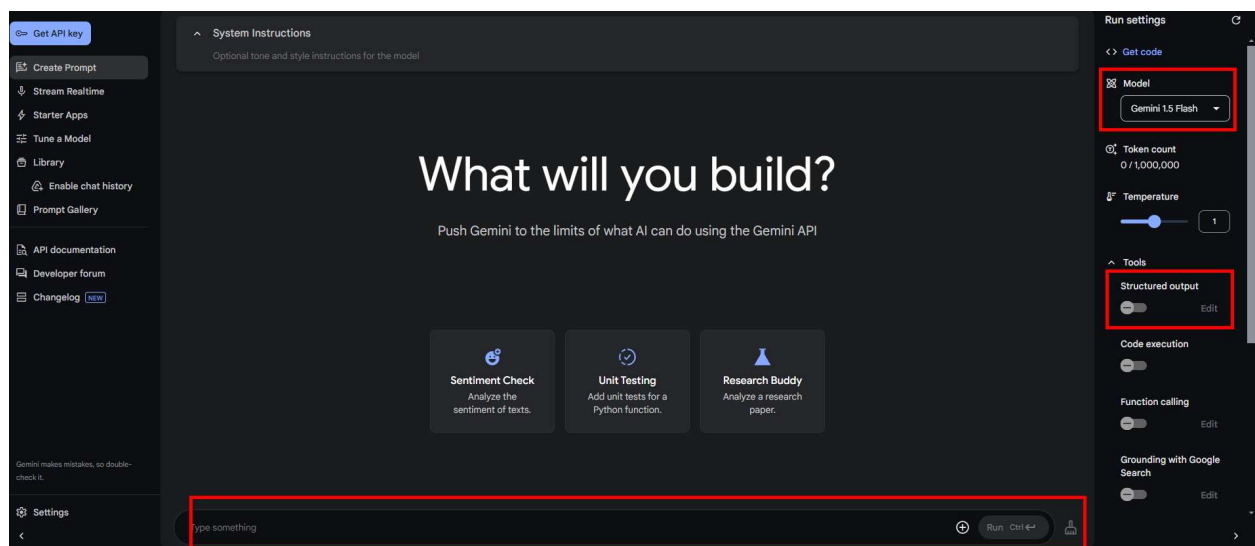
Support : ce sont des mini guide pour utiliser Gemini ou Mistral pour créer des agent IA, il ne suffira pas de faire un copier coller pour réussir le Hackathon, il faudra étoffer avec plus d'analyse de texte par l'agent pour réussir le Hackathon.

Gemini créer son agent IA :

Allez sur <https://aistudio.google.com/>



Créer une Clé API.



Sélectionner un modèle d'IA (Prendre Gemini 1.0 Flash), paramétrer une sortie en JSON, pensez à rajouter vos calculs de KPI et décrire le calcul dans le prompt pour chaque KPI.

Structured output schema

Enter an [OpenAPI schema object](#) to constrain the model output. See the [API documentation](#) for examples.

Code Editor

Visual Editor

Property

Sentiment

string

[]

*

Property

Inconfort

number

[]

*

Property

Urgence

boolean

[]

*

Écrire le prompt avec les instructions de sortie et tester

Dans ce message quel est le sentiment général du message, le taux d'inconfort entre 0 et 100% et si c'est un problème urgent à régler oui ou non ? Le message : @ENGIEpartFR \n6 mois d'attente et tjs aucune réponse de l'agence engie home service grand marseille à Aubagne sur un devis, malgré des multiples relances. Je reste coincé à appeler la plate-forme tél et à entendre qu'on me rappelle\nPour le numéro 1 du s

{

"Inconfort": 95,

"Sentiment": "Négatif",

"Urgence": true

}

Use code with caution.

Json

👍

🗨

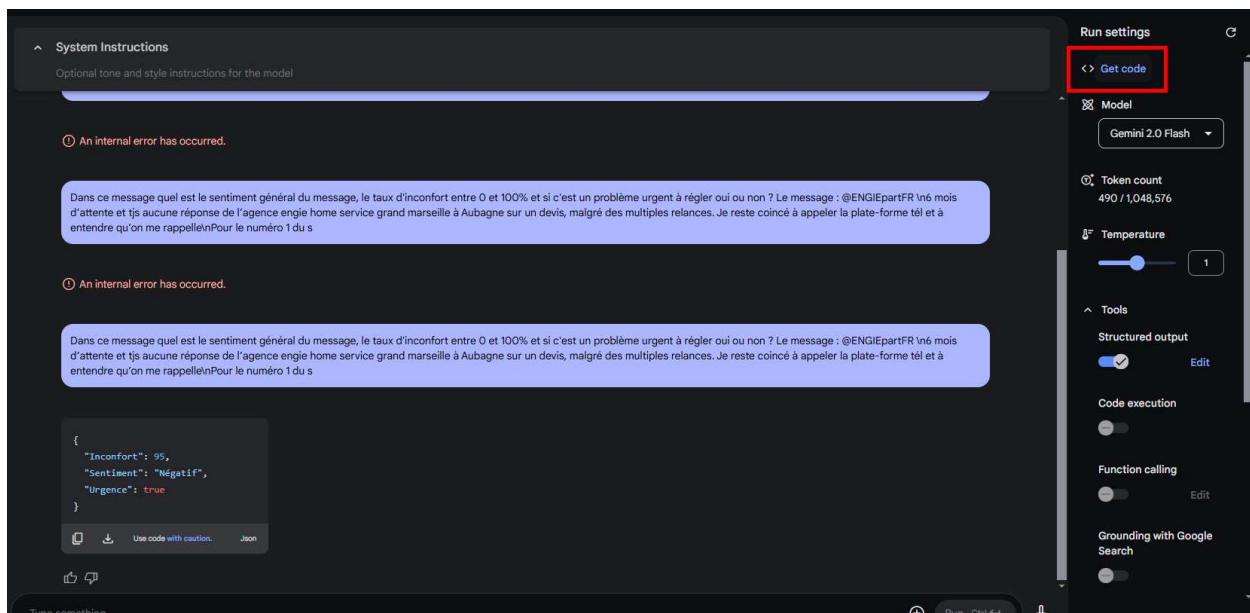
something

+

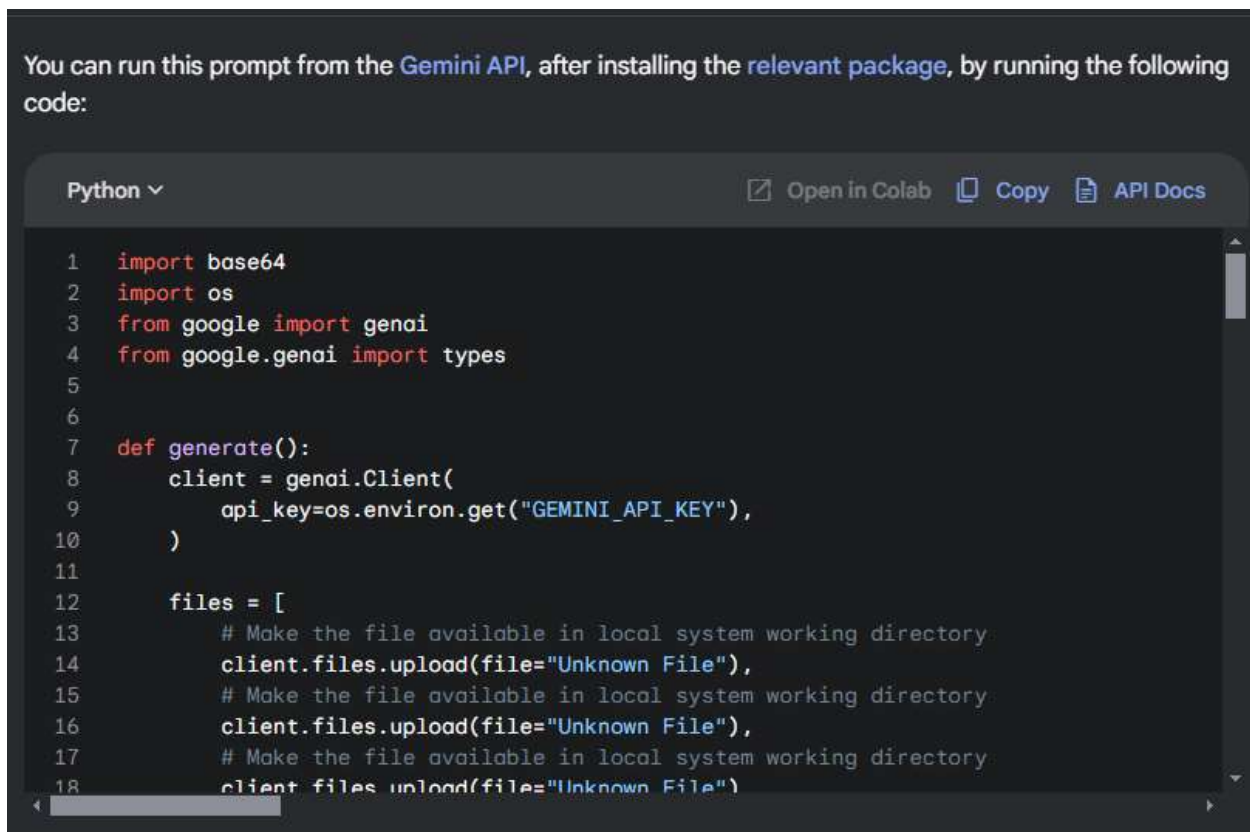
Run

Ctrl ↵

Récupérer le code à intégrer sur Python ici :



Corriger le code et l'intégrer pour calculer les KPI via Gemini :



Gemini peut traiter 3-4 messages par minutes grand maximum dans sa version gratuit.

Mistral :

Aller sur : <https://console.mistral.ai/home>

Commencez avec Mistral AI

Créer un agent

Utilisez des modèles, des instructions et des démonstrations pour créer un agent personnalisé à déployer sur le Chat.



Fine-tune un modèle

Créez et déployez un modèle fine-tuné à partir de vos propres données.



Créer un agent et activer le mode développeur.

< Agents Déployer

Untitled Agent
Aucune description

Modèle Mistral Nemo

Hasard Température 0,7

Instructions Prompt système

Je vais te donner des messages, tu devras évaluer l'inconfort entre 0 et 100%, le sentiment s'il est positif, neutre ou négatif et si le problème rencontré est Urgent ou non.

Démonstrations Few-shot prompts

Exemple 1

Exemple d'entrée

@ENGIEpartFR \n6 mois d'attente et tjs aucune réponse de l'agence engie home service grand marseille à Aubagne sur un devis, malgré des multiples relances. Je reste coincé à appeler la plateforme tél et à entendre qu'on me rappellenPour le numéro 1 du s

Exemple de sortie

```
{
  "Inconfort": 95,
  "Sentiment": "Négatif",
  "Urgence": true
}
```

Écrivez quelque chose pour tester l'agent avant de le déployer.

Untitled Ag... Demandez quelque chose...

Un agent se crée en mettant des instructions et une dizaine d'exemple. Rajouter dans le json le reste des KPI qu'il doit calculer (catégorie etc). Cliquer sur Déployer et tester en activant l'option Chat.

Déploiement

✓ API

Agent ID

✓ Le Chat

Essayer sur Le Chat

✓

Dernière mise à jour : 05/03/2025 15:16:40

Mettre à jour

Réinitialiser les changements

AC

@ENGIEpartFR \n6 mois d'attente et tjs aucune réponse de l'agence engie home service grand marseille à Aubagne sur un devis, malgré des multiples relances. Je reste coincé à appeler la plate-forme tél et à entendre qu'on me rappelle\nPour le numéro 1 du s

Effacer

M

Inconfort : 85%

Sentiment : Négatif

Urgence : Oui

Readapter le prompt et le mettre à jour si besoin.
 Pour l'appeler depuis python, copier l'Agent ID et créer une clé API ici :

Aperçu

REGISTRY

Agents

Modèles fine-tunés

Jeux de données

Batches

API

Clés API

Utilisation

Limites

CODESTRAL

Codestral

Vos clés API

Créer une nouvelle clé

Gérez vos clés API personnelles pour votre espace de travail actuel.

Vous pouvez créer jusqu'à 10 clés API et les supprimer lorsque vous n'en avez plus besoin.

Actives

Expirées

Vous n'avez pas encore de clé API. Générez-en une pour commencer à utiliser l'API.

Code en python pour appeler l'agent :

```
from mistralai import Mistral
client = Mistral(api_key="A insérer")
```

```
content = "Commentaire : " + content
response = client.agents.complete(
    agent_id=agent_id,
    messages=[{"role": "user", "content": content}]
)
response_text = response.choices[0].message.content.strip()
```