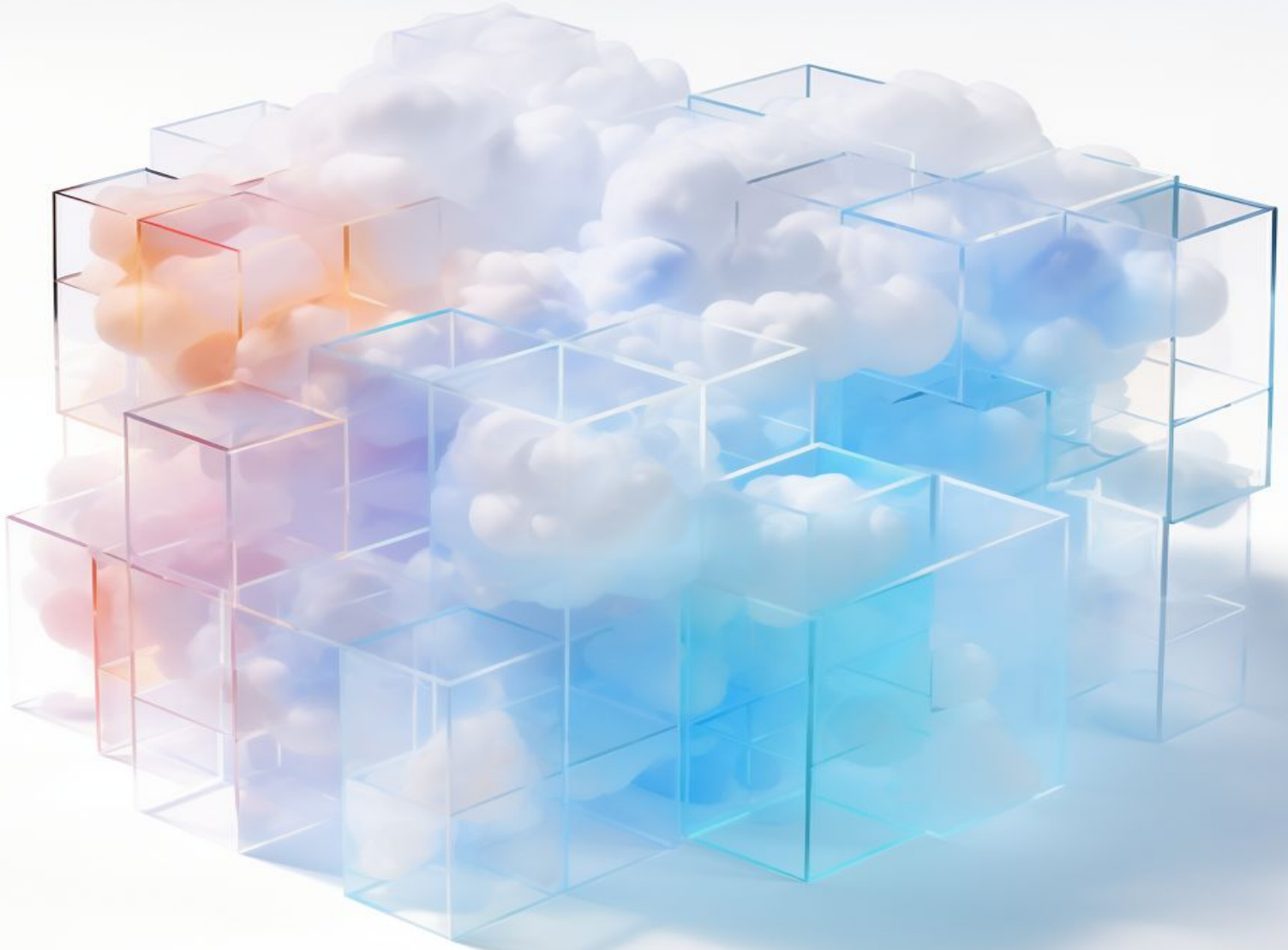


Déploiement d'un projet BigData / MLOps : Github API

Présentation

POC - Données Github

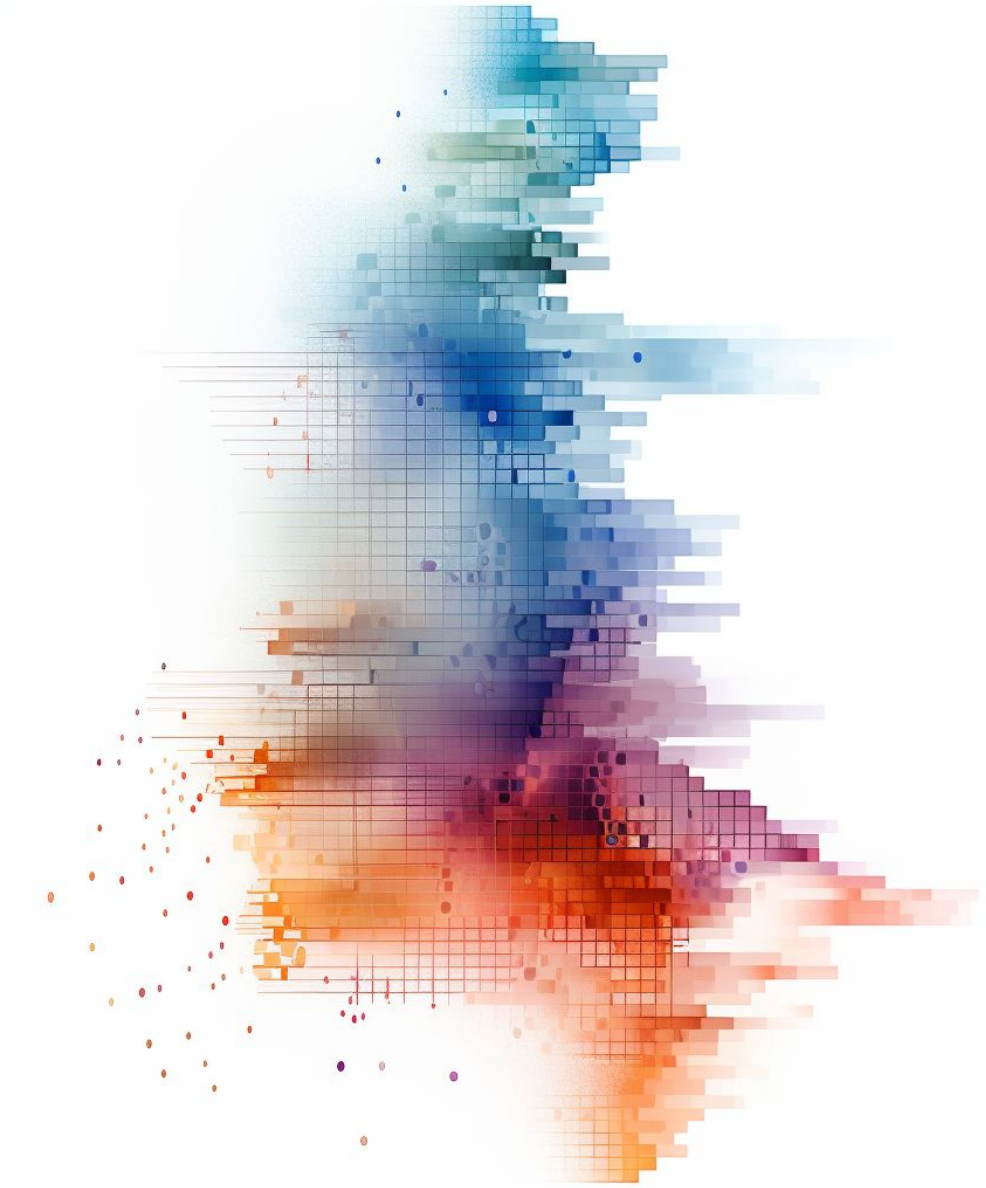
Mise en contexte:

Il est très fréquent de devoir réaliser des Proof-Of-Concept (POC) afin de démontrer la faisabilité et l'intérêt d'un développement.
Dans ce cas, on suppose qu'il s'agit de développer notre propre pipeline de traitement des données en temps-réel puis de valoriser ces données grâce à l'IA. Par ailleurs, cette pipeline d'ingestion/traitement/valorisation doit pouvoit être facilement déployée grâce aux techniques ML Ops : containerisation et orchestration des containers et des modèles.

Dans le cas de cette preuve de concept, il s'agit de travailler avec des données issues de l'API Github.

Quelques exemples d'applications possibles :

- Statistiques et trends prédictives sur les langages utilisés ;
- Statistiques et trends prédictives sur les mots-clés, et donc sur les types de projet ;
- Language models dédiées à l'aide à la prog sur certains types de projets ;
- Clustering de projets / topic model ;
- Analyse de sentiments sur les issues/reponses ;
- Prédiction de résolution d'issues en utilisant les issues et les request qui y répondent ;
- Prédiction de la croissance en terme d'usage d'un dépôt en analysant les tendances ;
- Recommandation de dépôts quand on est développeur d'un dépôt.



Présentation

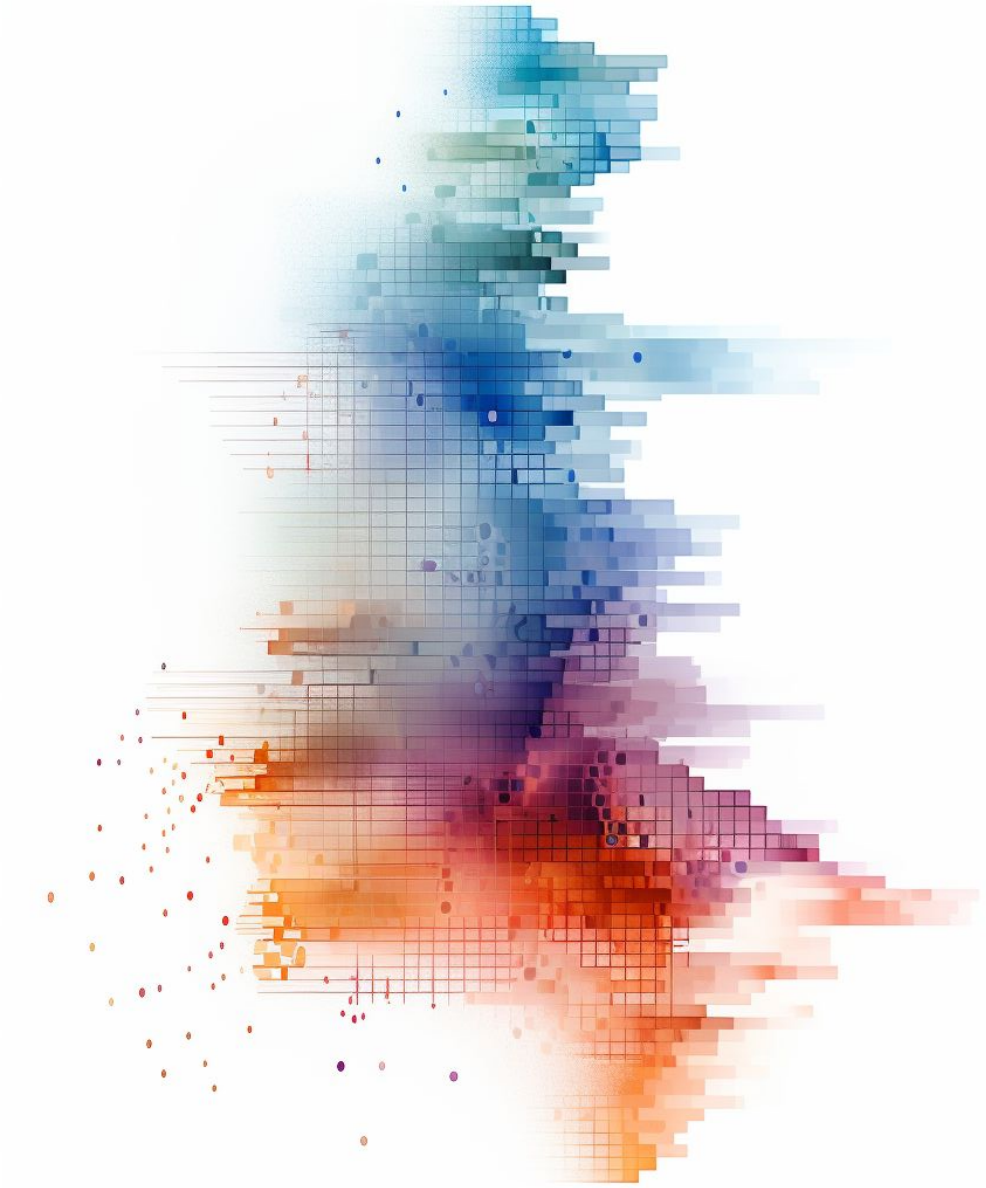
Livrables attendus (3 mars)

Synthèse au format A4
- Description des objectifs du projets
- Réalisations #1, #2 et #3
- Challenges et perspectives

Soutenance le 7 mars
- **Durée :** 15min + 5min de questions
- **Groupes :** 3 personnes
- Lors de la soutenance il faudra aborder les points de blocage ou saillant rencontrés, de même que les points techniques qui vous ont marqués et/ou que vous avez pu trouver ‘’amusant’’ lors de vos travaux.

Application
- Containers
- Minikube

L’objectif est que l’on puisse lancer votre application pour la tester.



Présentation

Step 1

Mettre en œuvre une solution d'ingestion et de visualisation temps-réel de données collectées via l'API

Il s'agit d'implémenter un pipeline de bout-en-bout de collecte, traitement et stockage de ces données issues de Github. Néanmoins, d'autres sources de données peuvent également servir à enrichir la collecte faite.

Pour restituer les données collectées, un tableau de synthèse (Dashboard) doit permettre de comparer les informations actuelles aux tendances des années précédentes via différentes visualisations qu'il vous faut imaginer.

Contraintes

- Mise à disposition des données sur une base **NoSQL** en sortie (MongoDB, ElasticSearch, etc)
- **Dashboard** de visualisation des données, par exemple avec MongoDB Charts ou Kibana
- **Kafka** et **Spark** dans le pipeline de gestion de données
- Containerisation

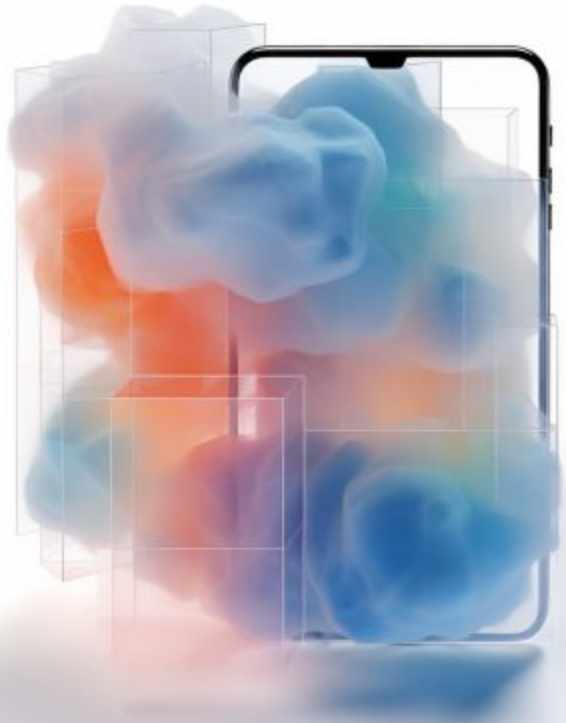
Présentation

Step 2

Proposer une solution valorisant les données recueillies, grâce à l'IA, à destination d'entreprises, d'associations ou tout autre type d'organisation ainsi que des particuliers.

Exemples:

- Statistiques et trends prédictives sur les langages utilisés pour guider les décideurs et les université sur les évolutions des langages ;
- Statistiques et trends prédictives sur les mots-clés, et donc sur les types de projet, par exemple pour des organismes type incubateurs d'entreprise ;
- Language models dédiées à l'aide à la programmation spécialisés sur certains types de projets ;
- Analyse de sentiments sur les issues/reponses pour gérer les priorités et aider les développeurs ;
- Prédiction de résolution d'issues en utilisant les issues et les request qui y répondent pour aider les dev ;
- Prédiction de la croissance en terme d'usage d'un dépôt en analysant les tendances, afin de guider les developpeurs ou les fondations soutenant l'open source à gérer leur modèle de croissance ;
- Recommandation de dépôts quand on est développeur d'un dépôt, afin de mieux connaitre l'existant et les concurrents ;
- etc.



Contraintes

- Déployer un modèle ML alimenté par les données
- Évaluer le modèle ML
- APIsation/Containerisation

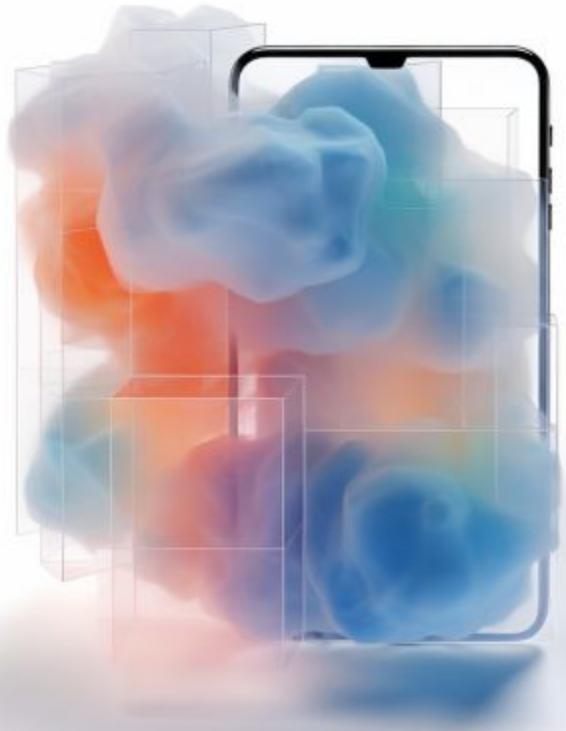
Présentation

Step 3

Proposer une solution d’évaluation et de monitoring de vos modèles ML

Il s’agit de vous assurer de la performance du ou des modèles déployés via, par exemple :

- une évaluation, qui peut être faite en continu ;
- des réapprentissages déclenchés automatiquement en fonction de l’évaluation ;
- un monitoring des données, pour évaluer leur évoluer et déclencher un réapprentissage ;
- l’apprentissage d’un modèle dit “streaming” capable d’ingérer des données temps-réel.



Contraintes

- Intégré à la pipeline
- Évaluer/réapprendre le modèle ML en continu
- Containerisation