**Plan d'attaque - Architecture du Projet Titanic Avancé**

# Architecture Globale :

## Infrastructure de Stockage des Données :

Utiliser une base de données distribuée comme Apache Cassandra pour gérer le volume massif de données.

Gestion du Volume de Données

Évolutivité Horizontale : Les bases de données distribuées peuvent être facilement étendues en ajoutant de nouveaux nœuds au cluster.

Haute Disponibilité : La distribution des données sur plusieurs nœuds améliore la disponibilité du système. En cas de défaillance d'un nœud, les données restent accessibles à partir d'autres nœuds, assurant ainsi la continuité du service.

Traitement Parallèle : Les bases de données distribuées peuvent effectuer des opérations en parallèle sur plusieurs nœuds, ce qui accélère le traitement des requêtes et des analyses, crucial dans un environnement Big Data où les performances sont une préoccupation majeure.

Gestion de Flux de Données : Avec l'arrivée continue de données dans un contexte de Big Data, une base de données distribuée peut traiter efficacement les flux de données en temps réel, offrant ainsi une capacité d'ingestion rapide.

Répartition de la Charge : La répartition de la charge sur plusieurs nœuds permet d'équilibrer la charge de travail, évitant ainsi les goulets d'étranglement et garantissant une utilisation efficace des ressources du système.

Support de Technologies Big Data : Certaines bases de données distribuées, telles qu'Apache Cassandra, HBase, ou Amazon DynamoDB, sont conçues spécifiquement pour traiter de grands ensembles de données distribuées et sont compatibles avec l'écosystème Big Data.

Configurer des connecteurs pour permettre l'ajout continu de données au fur et à mesure qu'elles arrivent.

## Traitement des Données :

Mettre en place un pipeline de traitement de données en temps réel ou par lots, selon la fréquence d'arrivée des données.

Utiliser Apache Spark pour le traitement par lots et Kafka pour le traitement en temps réel.

Implémenter des mécanismes de nettoyage et de prétraitement des données pour garantir la qualité.

## Stockage de Modèles et Résultats :

Utiliser un système de stockage distribué pour sauvegarder les modèles entraînés et les résultats de prédiction.

Considérer l'utilisation de technologies comme Hadoop Distributed File System (HDFS).

# B. Rôles et Responsabilités :

## Data Architect :

Conception de la structure de la base de données distribuée.

Configuration des connecteurs pour l'ingestion continue de données.

Gestion de l'infrastructure de stockage.

## Data Engineers :

Développement et maintenance des pipelines de traitement de données.

Mise en place de mécanismes d'évolutivité pour faire face à l'augmentation des données.

## Data Scientists :

Développement des modèles prédictifs en utilisant des bibliothèques telles que Scikit-Learn, TensorFlow ou PyTorch.

Intégration des modèles dans le pipeline de traitement des données.

# Organisation du Projet GitHub :

## Créer le Projet GitHub :

Créer un nouveau dépôt sur GitHub pour le projet Titanic avancé.

## Readme du Projet :

Inclure les noms et rôles de chaque membre de l'équipe.

Ajouter une brève description du projet et des objectifs.

## Arborescence des Fichiers :

Notebook : Pour les fichiers .ipynb.

Data : Pour stocker les ensembles de données.

Docs : Pour les présentations (ex. PowerPoint) et autres documents.

Src : Pour les scripts Python (.py).

Test : Pour les fichiers de test.

### Présentation de l'Architecture :

Chaque membre présente son rôle et responsabilités.

Discussion sur l'architecture globale et les choix technologiques.

Définition des étapes de développement et de déploiement.

### Délais et Livrables en une Journée :

#### Journée 1 :

Configuration initiale du dépôt GitHub.

Création de l'arborescence des fichiers.

Mise en place de la base de données distribuée (par exemple, Apache Cassandra).

Développement initial du pipeline de traitement des données.

### Prochaines Étapes :

##### Jours Suivants :

Développement des modèles prédictifs.

Intégration des modèles dans le pipeline.

Tests et optimisation.

Préparation de la présentation et des documents.

### Commencer à Coder :

Les membres commencent à coder selon les rôles et responsabilités définis.

Communication continue au sein de l'équipe pour assurer une collaboration efficace.