Migration Pandas → PySpark & industrialisation

Contexte du projet

Votre client dispose aujourd'hui d'un code de traitement de données écrit en Pandas. Les volumes de données augmentant, les temps de traitement deviennent trop longs et pénalisent les usages métiers.

Votre mission est de proposer et mettre en œuvre une solution permettant d'accélérer ces traitements, en particulier via une migration vers PySpark. Au-delà de la simple traduction, l'objectif est de professionnaliser la chaîne de développement :

- code migré de Pandas vers PySpark,
- tests unitaires garantissant que les résultats PySpark sont identiques à ceux de Pandas,
- mise en place d'un système d'automatisation (pre-commit) pour contrôler la qualité du code avant chaque commit.

Le projet devra être mené en suivant une démarche Agile, avec gestion des tâches par sprints et suivi via Trello (ou autre outils).

Objectifs : À l'issue du projet, vous devrez être capable de :

1. Comprendre les différences entre Pandas et PySpark (modèle distribué, performances, API).
2. Installer et prendre en main Spark / PySpark en local (et/ou via Docker) puis sur Databricks.
3. Migrer du code métier existant de Pandas vers PySpark.
4. Mettre en place des tests unitaires pour valider la migration (mêmes entrées → mêmes sorties).
5. Configurer un hook pre-commit pour automatiser des contrôles (formatage, linting, tests).
6. Organiser votre travail en mode Agile (sprints, backlog, Trello, revues).

Modalités pédagogiques

Durée : 3 jours

Travail : individuel

Méthode de gestion de projet : Agile (simplifiée pour commencer)

Organisation du travail (sprints & planning)

Agile & Trello

Jour 1 (matin) – ~1 h

- Vous documenter sur les principes de base de la méthode Agile.
- Créer un tableau Trello (ou équivalent) et structurer vos tâches :
- Backlog / À faire / En cours / Fait

- 1 carte = 1 tâche claire (ex: Installer Spark en local, Migrer le script X)

Vous devrez actualiser votre Trello à la fin de chaque sprint.

Sprint 1 – Découverte Spark & PySpark

Jour 1 (matin et après-midi) – ~4 h

- Veille technique sur Spark / PySpark : concepts clés (RDD, DataFrames, cluster, driver, etc.).
- Installation de Spark en local : en natif (Spark + Python + Java), et/ou via Docker.
- Prrise en main de PySpark : création de SparkSession, quelques transformations simples (DataFrames ou RDD).

Sprint 2 – Entraînement PySpark

Jour 1 (après-midi) – ~2h

- S'exercer sur des fonctions basiques dont vous aurez besoin pour la migration : sélections, filtres, agrégations, joins, groupBy, etc.
- Reproduire des traitements simples d'un document d'exemples (fourni) en version PySpark.
- Documenter ces essais dans un notebook ou des scripts séparés.

Sprint 3 – Migration Pandas → PySpark

Jour 2 – ~1 jour

- Analyser le code Pandas existant (dans src/pandas/).
- Traduire chaque fichier en version PySpark (dans src/pyspark/).
- S'assurer que l'API et la logique métier restent identiques (mêmes entrées / mêmes sorties).

Sprint 4 – Tests unitaires & pre-commit + présentation

Jour 3 – ~1 jour

- Mise en place des tests unitaires
- Mise en place de pre-commit
- Présentation (10 minutes) : Démo, Bilan : difficultés, limites, pistes d'amélioration.

Modalités d'évaluation

Repository GitHub Presentation orale

Livrables

Repository GitHub Présentation orale

Critères de performance

Utilisation correcte de l'API PySpark Présence de tests couvrant les principales fonctions migrées. Mise en place effective de pre-commit. Mise à jour des tâches et des sprints.