Analyse exploratoire de données criminelles avec Azure et Databricks

Sankara Kabem Abdoul Charif

14 juillet 2025

Objectif

Ce projet vise à démontrer la mise en place d'un pipeline cloud complet pour l'analyse exploratoire (EDA) d'un jeu de données volumineux (**8 millions de lignes**), en utilisant les services **Azure** et le traitement distribué avec **Apache Spark via Databricks**.

Architecture du projet

Google Drive (CSV brut)
↓
Azure Data Factory (source HTTP)
↓
Azure Blob Storage (fichier stocké)
↓
Azure Databricks (cluster Spark)
↓
Notebook PySpark (traitement & visualisation)

Étapes réalisées

1. Préparation des données

Le fichier `Crimes.csv` contenant des infractions (type, lieu, date, etc.) a été placé sur Google Drive. Il contient environ 8 millions de lignes.

2. Création des ressources Azure

- Groupe de ressources dédié
- Azure Data Factory (ADF)
- Compte de stockage Azure (`crimestockage28`)
- Azure Databricks (cluster Spark 3.x)

3. Ingestion avec Azure Data Factory

- Source : connecteur HTTP (Google Drive)
- Destination : Azure Blob Storage (`Crimes.csv`)
- Résultat : pipeline actif transférant automatiquement les données

4. Traitement avec Azure Databricks

- Connexion du notebook au cluster
- Chargement du fichier via PySpark :

```
spark.conf.set("fs.azure.account.key.crimestockage28.blob.core.windows.net", "<clé>
df = spark.read.format("csv").option("header", "true") \
     .option("inferSchema", "true") \
     .load("wasbs://crimesdatabrutes@crimestockage28.blob.core.windows.net/Crimes.c
```

5. Nettoyage et transformation

- Renommage des colonnes (francisation)
- Conversion des dates en format `timestamp`
- Suppression des valeurs nulles et doublons
- Création de vues SQL temporaires

6. Analyse exploratoire (EDA)

- Statistiques descriptives (Spark SQL)
- Requêtes groupées : infractions, arrestations, années
- Visualisations : top 10 types d'infractions, distribution temporelle

Technologies utilisées

Outil	**Usage**
Google Drive	Hébergement initial du fichier
Azure Data Factory	Ingestion automatisée (HTTP → Blob)
Azure Blob Storage	Stockage des fichiers
Azure Databricks	Traitement et visualisation
Apache Spark	Traitement distribué des données
PySpark	Nettoyage et transformation
Matplotlib / Seaborn	Visualisation graphique

Résultats obtenus

- Chargement et traitement d'un fichier de 8M lignes avec Spark
- Analyse des types d'infractions les plus fréquents
- Pipeline cloud reproductible, scalable et automatisé

Perspectives

- Construction d'un entrepôt de données (modèle en étoile)
- Export vers Power BI pour visualisation métier
- Ajout d'indicateurs de performance (KPI)

Liens utiles

- Données : Lien Google Drive
- Notebook Databricks : *à exporter au format .ipynb*
- Repository GitHub : Lien Google Drive