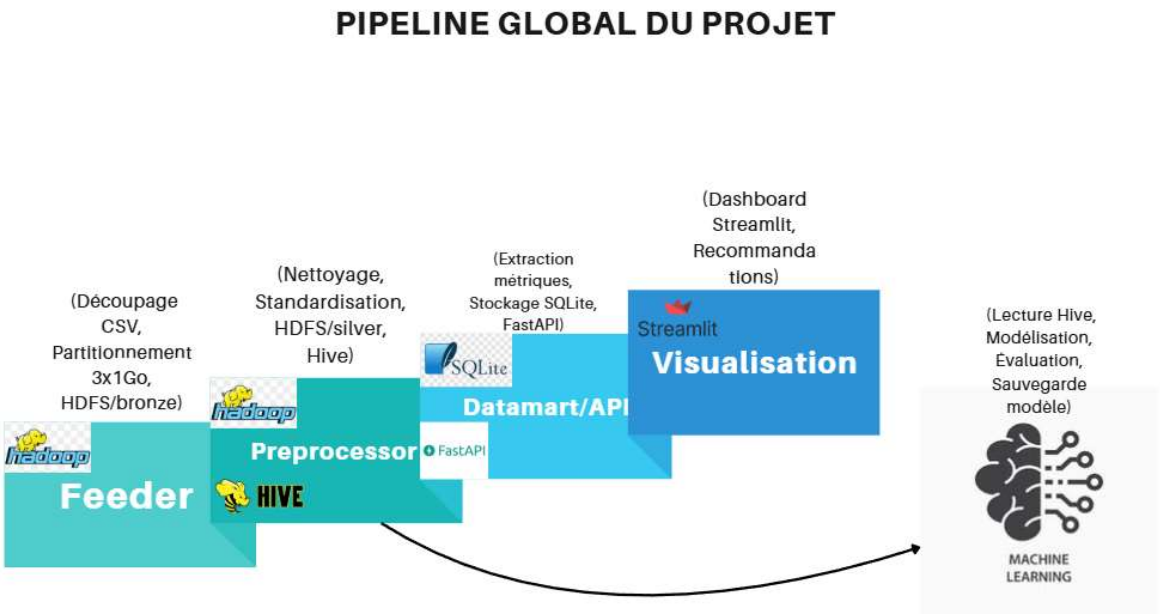
Rapport Projet Final – US Accidents

1. Architecture du code, classes/fonctions réutilisables, logger

Notre projet est structuré en modules clairs :
- feeder/ : ingestion et découpage des données
- preprocessor/ : nettoyage et standardisation
- ml_training/ : machine learning
- datamart/ : API et visualisation

Chaque module utilise des classes et fonctions réutilisables pour faciliter la maintenance.
Un logger personnalisé (`get_logger`) est utilisé dans chaque étape pour tracer les opérations et erreurs.



2. Partie Feeder – Clarté, data management, formats

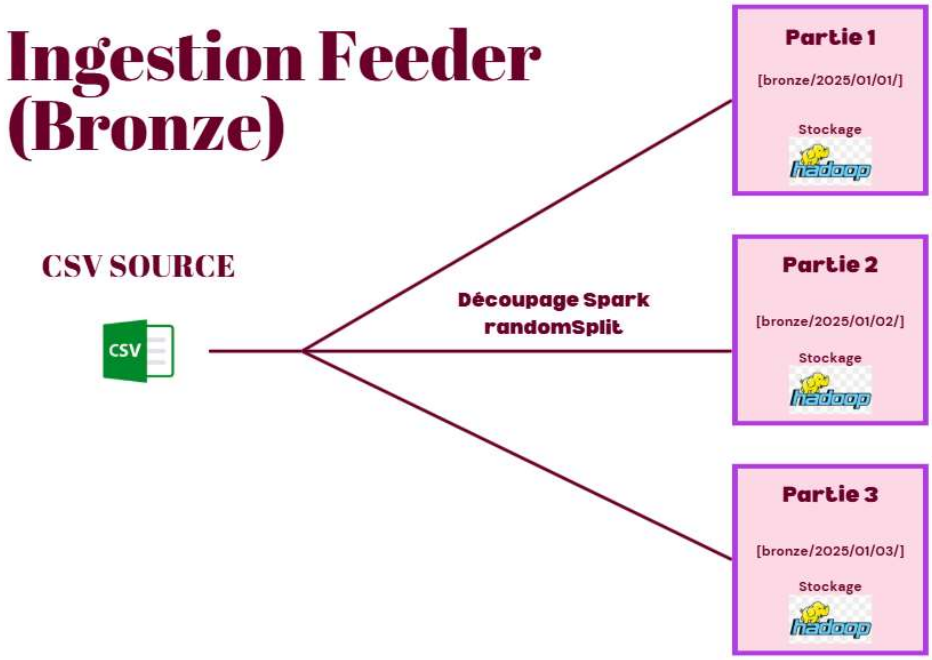
Le feeder simule une ingestion batch incrémentale typique d'un data lake. Le fichier CSV source est découpé en trois parties équivalentes à l'aide de Spark, car sinon le fichier était trop volumineux.

Chaque partie est injectée dans la couche bronze, qui est stockée dans HDFS (Hadoop Distributed File System), à une date simulée différente.

Chaque jour, le dossier cible contient l'historique complet jusqu'à la date du jour, grâce à l'utilisation du mode overwrite.

Le format Parquet a été choisi pour ses avantages en termes de performance, de compression et de gestion de schéma.

Cette approche respecte les bonnes pratiques de data management :
- Découpage équitable et reproductible
- Stockage distribué dans HDFS
- Stockage partitionné par date
- Utilisation d'un format optimisé pour le big data

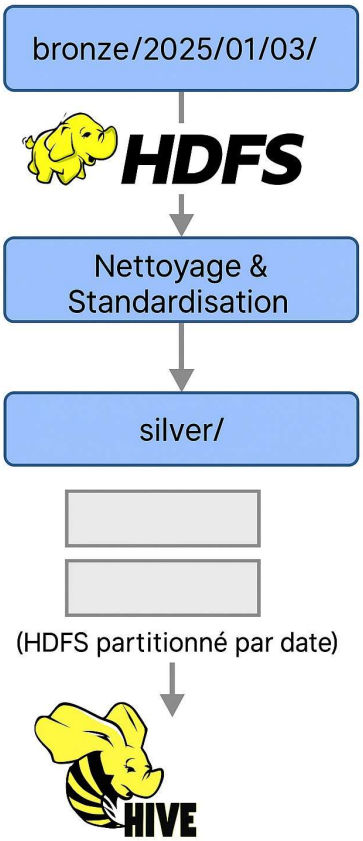


Voici un résumé de notre code mettant en avant sa logique :

```
df = spark.read.option("header", True).option("inferSchema", True) \
    .csv("hdfs://namenode:9000/source/raw_us_accidents.csv")

for col in df.columns:
    if any(c in col for c in [' ', '(', ')', ';', '{', '}', '=', '\n', '\t']):
        df = df.withColumnRenamed(col, col.replace(' ', '_')
                                         .replace('(', '')
                                         .replace(')', '')
                                         .replace('/', '_'))

part1, _, _ = df.randomSplit([0.33, 0.33, 0.34], seed=42)

output_path = "hdfs://namenode:9000/bronze/2025/01/01/"
part1.write.mode("overwrite").parquet(output_path)
```

3. Partie Preprocessor – Clarté et détails

Le preprocessor lit la dernière partition bronze, applique un nettoyage (types, minuscules, suppression des valeurs aberrantes) et standardise les données.
Les données sont ensuite écrites dans la couche silver, partitionnées par date, prêtes pour l'analyse ou le ML.
L'intégration Hive permet un accès SQL et une gestion centralisée des schémas.

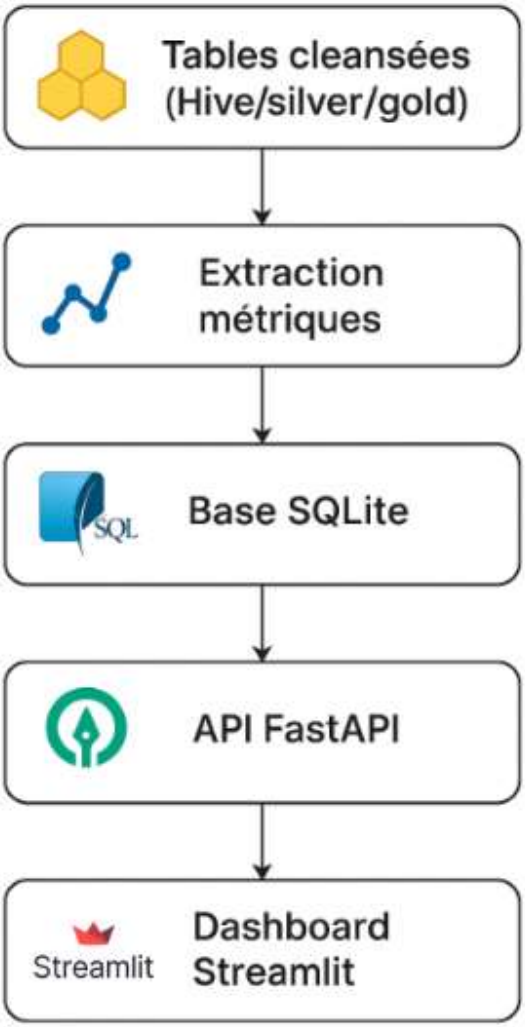


Voici un résumé de notre code mettant en avant sa logique :

```
cols_to_keep = [
    "Start_Time", "Start_Lat", "Start_Lng", "Severity",
    "Description", "City", "State"
]
df_filtered = df.select(*cols_to_keep)

df_cleaned = df_filtered \
    .withColumn("Start_Time", to_timestamp("Start_Time")) \
    .withColumn("Start_Lat", col("Start_Lat").cast("double")) \
    .withColumn("Start_Lng", col("Start_Lng").cast("double")) \
    .withColumn("Severity", col("Severity").cast("int")) \
    .withColumn("Description", lower(trim(col("Description")))) \
    .withColumn("City", lower(trim(col("City")))) \
    .withColumn("State", lower(trim(col("State")))) \
    .withColumn("year", year("Start_Time")) \
    .withColumn("month", month("Start_Time")) \
    .withColumn("day", dayofmonth("Start_Time"))
```

4. Partie Datamart – Clarté et détails

Le datamart extrait des métriques business pertinentes (par exemple : accidents par ville, par sévérité, etc.) depuis la couche silver/gold.
Les résultats sont stockés dans une base SQLite pour une consultation rapide via API ou dashboard.
L'API FastAPI expose ces métriques de façon paginée et documentée.



5. Partie ML – Clarté et détails

Le module ML lit les données cleansées, sélectionne les features pertinentes, et entraîne un modèle Random Forest pour prédire la sévérité des accidents.
L'évaluation (accuracy, matrice de confusion) est sauvegardée et visualisée.
Le pipeline est automatisé et reproductible.

6. Déploiement sur Yarn, choix des paramètres Spark

Les jobs Spark sont soumis sur Yarn pour bénéficier du parallélisme.
Nombre d'exécuteurs, mémoire, etc. sont choisis selon la taille des données et la RAM disponible.

Exemple de commande Spark submit :
`spark-submit --master yarn --num-executors 4 --executor-memory 4G ...`

Capture d'écran de Yarn:

Worker Id	Address	State	Cores	Memory	Resources
worker-20250625010855-172.19.0.12-39673	172.19.0.12:39673	ALIVE	14 (0 Used)	12.0 GiB (0.0 B Used)	

Running Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration

Completed Applications (17)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
app-20250627195449-0016	Gold Writer	14	8.0 GiB		2025/06/27 19:54:49	root	FINISHED	1.1 min
app-20250627194055-0015	Gold Writer	14	8.0 GiB		2025/06/27 19:40:55	root	FINISHED	1.2 min
app-20250627181355-0014	Gold Writer	14	8.0 GiB		2025/06/27 18:13:55	root	FINISHED	1.2 min
app-20250627180905-0013	Gold Writer	14	8.0 GiB		2025/06/27 18:09:05	root	FINISHED	1.2 min
app-20250627180555-0012	Gold Writer	14	8.0 GiB		2025/06/27 18:05:55	root	FINISHED	1 s
app-20250626223410-0011	HiveWriter	14	8.0 GiB		2025/06/26 22:34:10	root	FINISHED	5.3 min
app-20250626222216-0010	HiveWriter	14	8.0 GiB		2025/06/26 22:22:16	root	FINISHED	7.3 min
app-20250626220901-0009	HiveWriter	14	1024.0 MiB		2025/06/26 22:09:01	root	FINISHED	4.4 min
app-20250626215807-0008	HiveWriter	14	1024.0 MiB		2025/06/26 21:58:07	root	FINISHED	6.9 min
app-20250626214832-0007	HiveWriter	14	1024.0 MiB		2025/06/26 21:48:32	root	FINISHED	2 s
app-20250626213836-0006	HiveWriter	14	1024.0 MiB		2025/06/26 21:38:36	root	FINISHED	6.2 min
app-20250626213207-0005	Cleaner	14	1024.0 MiB		2025/06/26 21:32:07	root	FINISHED	5.2 min
app-20250626213115-0004	Cleaner	14	1024.0 MiB		2025/06/26 21:31:15	root	FINISHED	22 s
app-20250626212021-0003	HiveWriter	14	1024.0 MiB		2025/06/26 21:20:21	root	FINISHED	11 min
app-20250626204014-0002	Cleaner	14	1024.0 MiB		2025/06/26 20:40:14	root	FINISHED	5.2 min

7. Problématiques et Business Value

- Problématique : Prédire la sévérité des accidents pour mieux cibler les actions de prévention.
- Business Value :
  - Optimisation des ressources d'intervention
  - Identification des zones à risque
  - Aide à la prise de décision pour les collectivités

8. API

L'API développée avec FastAPI permet d'accéder facilement aux métriques et aux datamarts.
Elle propose des endpoints documentés, paginés, et peut être sécurisée selon les besoins.
Cela facilite l'intégration avec d'autres outils ou applications métiers.

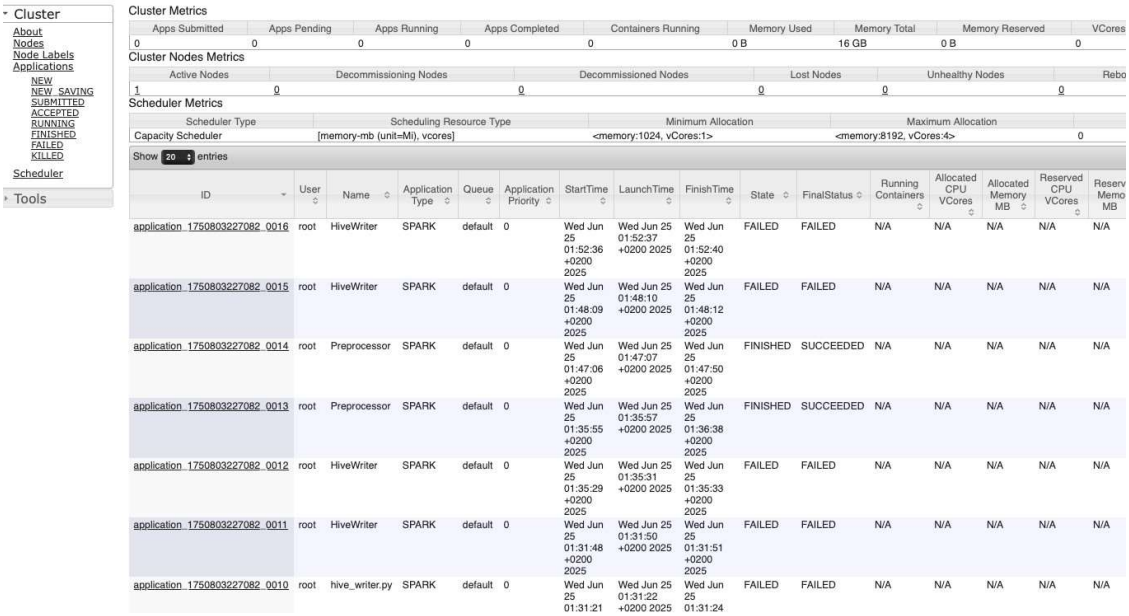
9. Visualisation et recommandations

Un dashboard Streamlit présente :

- L'accuracy du modèle
- La matrice de confusion

10. Spark UI, database, Hive

Capture d'écran de notre interface Spark UI :



Cluster

About
Nodes
Node Labels
Applications
NEW
NEW_SAVING
SUBMITTED
ACCEPTED
RUNNING
FINISHED
FAILED
KILLED
Scheduler

Tools

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores
0	0	0	0	0	0 B	16 GB	0 B	0

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebo
1	0	0	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:1024, vCores:1>	<memory:8192, vCores:4>	0

Show 20 entries

ID	User	Name	Application Type	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCores	Allocated Memory MB	Reserved CPU VCores	Reserv Memo MB
application_1750803227082_0016	root	HiveWriter	SPARK	default	0	Wed Jun 25 01:52:36 +0200 2025	Wed Jun 25 01:52:37 +0200 2025	Wed Jun 25 01:52:40 +0200 2025	FAILED	FAILED	N/A	N/A	N/A	N/A	N/A
application_1750803227082_0015	root	HiveWriter	SPARK	default	0	Wed Jun 25 01:48:09 +0200 2025	Wed Jun 25 01:48:10 +0200 2025	Wed Jun 25 01:48:12 +0200 2025	FAILED	FAILED	N/A	N/A	N/A	N/A	N/A
application_1750803227082_0014	root	Preprocessor	SPARK	default	0	Wed Jun 25 01:47:06 +0200 2025	Wed Jun 25 01:47:07 +0200 2025	Wed Jun 25 01:47:50 +0200 2025	FINISHED	SUCCEEDED	N/A	N/A	N/A	N/A	N/A
application_1750803227082_0013	root	Preprocessor	SPARK	default	0	Wed Jun 25 01:35:55 +0200 2025	Wed Jun 25 01:35:57 +0200 2025	Wed Jun 25 01:36:38 +0200 2025	FINISHED	SUCCEEDED	N/A	N/A	N/A	N/A	N/A
application_1750803227082_0012	root	HiveWriter	SPARK	default	0	Wed Jun 25 01:35:29 +0200 2025	Wed Jun 25 01:35:31 +0200 2025	Wed Jun 25 01:35:33 +0200 2025	FAILED	FAILED	N/A	N/A	N/A	N/A	N/A
application_1750803227082_0011	root	HiveWriter	SPARK	default	0	Wed Jun 25 01:31:48 +0200 2025	Wed Jun 25 01:31:50 +0200 2025	Wed Jun 25 01:31:51 +0200 2025	FAILED	FAILED	N/A	N/A	N/A	N/A	N/A
application_1750803227082_0010	root	hive_writer.py	SPARK	default	0	Wed Jun 25 01:31:21	Wed Jun 25 01:31:22 +0200 2025	Wed Jun 25 01:31:24	FAILED	FAILED	N/A	N/A	N/A	N/A	N/A

11. Optimisations

- Utilisation du format Parquet et du partitionnement
- Repartitionnement des données pour équilibrer la charge
- Indexation des variables catégorielles
- (Optionnel) Optimisation des hyperparamètres du modèle ML

Business Plan – Prédiction de la gravité des accidents de la route aux États-Unis

1. Résumé Exécutif

Ce projet vise à développer une solution innovante permettant de prédire la gravité des accidents de la route aux États-Unis en s'appuyant sur des données météorologiques, temporelles et géographiques. L'objectif est d'optimiser les dispositifs de prévention et d'intervention des autorités locales, afin de réduire le nombre de victimes et d'améliorer la gestion des ressources d'urgence.

2. Problématique

Comment prédire la gravité des accidents de la route aux États-Unis à partir des conditions météorologiques, temporelles et géographiques, afin d'optimiser les dispositifs de prévention et d'intervention des autorités locales ?

3. Opportunité

- Les accidents graves représentent un enjeu humain et financier majeur.
- Les autorités locales disposent de données massives mais peu exploitées pour la prédiction et l'anticipation.
- L'analyse croisée des facteurs météo, temporels et géographiques ouvre la voie à une prévention ciblée et à une meilleure allocation des secours.

4. Solution Proposée

- Ingestion automatisée de données d'accidents enrichies par des données météo, temporelles et géographiques, stockées dans un data lake (HDFS).
- Nettoyage et standardisation des données pour garantir leur qualité.
- Modélisation prédictive (Random Forest ou autre) pour estimer la gravité probable d'un accident selon les conditions observées.
- Datamart et API pour exposer les prédictions et indicateurs clés aux autorités et partenaires.
- Dashboard interactif pour visualiser les zones à risque, les tendances et les recommandations d'action.

5. Valeur Ajoutée

- Pour les autorités locales :
  - Anticipation des accidents graves selon la météo, l'heure, la localisation

  - Déploiement proactif des secours et des dispositifs de prévention
  - Optimisation des campagnes de sensibilisation et de la gestion du trafic

- Pour les assureurs et acteurs privés :
  - Meilleure évaluation du risque
  - Affinage de la tarification et des offres

- Pour les citoyens :
  - Routes plus sûres
  - Réduction du nombre et de la gravité des accidents

6. Marché Cible

- Collectivités locales, services de sécurité routière, services d'urgence
- Compagnies d'assurance
- Entreprises de transport/logistique
- Startups de la mobilité intelligente

7. Modèle Économique

- Vente de licences logicielles ou abonnement SaaS pour l'accès à la plateforme prédictive
- Abonnement à l'API de prédiction et aux dashboards
- Prestations de conseil et d'intégration
- Formation à l'utilisation de la solution

8. Plan de Déploiement

1. Phase pilote : Déploiement sur une grande ville ou un État, intégration des données locales
2. Industrialisation : Extension à l'échelle nationale, enrichissement des sources de données (météo temps réel, trafic…)
3. Évolution : Ajout de modules d'IA avancée, recommandations automatiques, alertes en temps réel

9. Indicateurs de Succès

- Réduction du nombre d'accidents graves sur les zones ciblées
- Diminution du temps d'intervention des secours
- Adoption de la solution par X collectivités/assureurs
- Taux d'utilisation de l'API et du dashboard

10. Risques et Atténuations

- Qualité et disponibilité des données : Mise en place de contrôles et enrichissement continu
- Adoption par les utilisateurs : Accompagnement au changement, démonstration de la valeur ajoutée
- Scalabilité : Architecture cloud-native, tests de montée en charge

11. Conclusion

Ce projet s'inscrit dans une démarche de smart city et de sécurité routière augmentée par la donnée. Il combine la puissance du Big Data, du Machine Learning et de la visualisation pour générer un impact concret sur la sécurité, l'efficacité opérationnelle et la prise de décision des autorités locales.