

Big Data et machine learning

2024-2025



réalisée par:

ELKADIRI ZOUHRA

1.Introduction

Dans ce projet, l'objectif principal était de découvrir et d'utiliser des outils de traitement de données tels qu'Apache Nifi, Apache Spark et Python pour traiter et visualiser des logs générés par un site e-commerce. Ces logs, qui enregistrent les événements utilisateurs, ont été utilisés pour créer un tableau de bord analytique, afin d'aider à la prise de décision. Ce projet est structuré en plusieurs étapes, allant de la collecte des logs à la création du tableau de bord final.

2. Objectifs du Projet

- Utilisation d'Apache Nifi : Préparer les données des logs en les récupérant et les transférant vers un répertoire spécifique.
- Traitement des données avec Apache Spark : Agréger les données des logs à l'échelle horaire, en calculant des métriques (comme la somme des prix des ventes par produit).
- Création du Dashboard : Développer une interface Python permettant de visualiser les données agrégées sur une période donnée.

3. Outils Utilisés

- Apache Nifi : Un outil de gestion de flux de données qui permet d'automatiser le processus de collecte, transformation et transfert de données.
- Apache Spark : Un moteur de traitement de données en parallèle permettant de traiter de grandes quantités de données rapidement.
- Python : Langage de programmation utilisé pour le script du tableau de bord afin de visualiser les résultats dans un format facile à comprendre.

4. Description du Projet

4.1. Description de site e-commerce:

Le projet consiste à simuler un site e-commerce dédié à la vente de chocolat, où les utilisateurs peuvent acheter une variété de produits chocolatés. Ce site génère des logs à chaque action effectuée par un utilisateur, comme les achats.

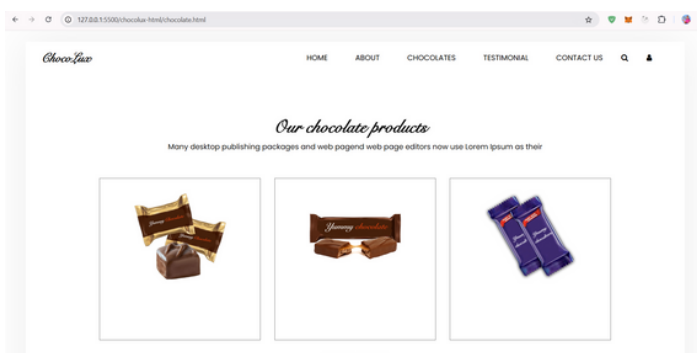


figure1: le site e-commerce ChocoLux

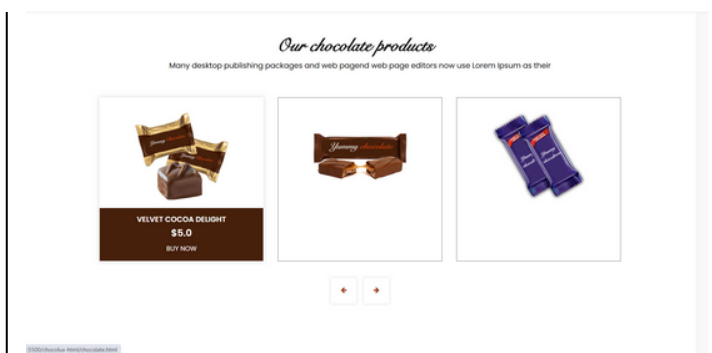


figure2: l'action d'achat dans le site e-commerce ChocoLux

4.2.Objectif Principal

L'objectif principal du projet est de traiter les logs générés par le site e-commerce, d'en extraire des informations pertinentes pour les agréger et de créer un tableau de bord permettant une analyse détaillée des ventes de chocolat. Ce tableau de bord vise à faciliter la prise de décisions commerciales en offrant des insights sur les produits les plus populaires, les tendances de consommation, et d'autres métriques clés telles que le montant total des ventes.

5.Étapes de Réalisation

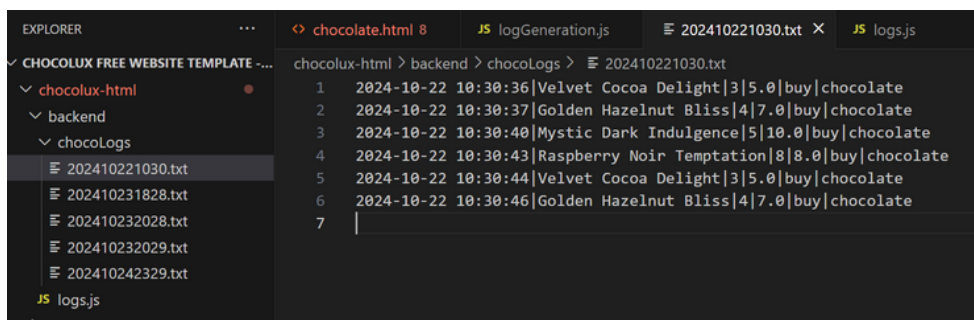
5.1. Génération des logs

Dans cette partie du projet, nous allons nous concentrer sur la génération des logs chaque fois qu'un utilisateur effectue une action importante, comme acheter un produit. L'objectif est de capturer ces événements et de les enregistrer dans des fichiers log pour une analyse ultérieure. Cette génération de logs est réalisée à l'aide de Node.js et de listeners d'événements qui surveillent les actions des utilisateurs, telles que l'achat de produits.

• Utilisation de Node.js pour la Génération des Logs

Node.js est utilisé pour gérer le backend de l'application e-commerce et pour enregistrer les logs des actions des utilisateurs. L'une des actions les plus importantes dans le contexte de ce projet est l'achat d'un produit. Chaque fois qu'un utilisateur clique sur le bouton d'achat ("Buy") d'un produit, un événement est déclenché, et un log est généré pour enregistrer cet événement. Ce log contiendra des informations telles que :

- La date et l'heure de l'achat
- Le nom de produit
- L'ID du produit acheté
- Le prix du produit
- L'action qu'est "buy"
- La page ou la section du site d'où provient cette action



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders 'chocolux-html', 'backend', and 'chocoLogs'. Under 'chocoLogs', several log files are listed, including '202410221030.txt'. The code editor shows the content of '202410221030.txt', which contains a list of log entries. Each entry is a string representing a log record, with fields separated by pipes. The fields include a timestamp, a product name, a price, and an action.

```
1 2024-10-22 10:30:36|Velvet Cocoa Delight|3|5.0|buy|chocolate
2 2024-10-22 10:30:37|Golden Hazelnut Bliss|4|7.0|buy|chocolate
3 2024-10-22 10:30:40|Mystic Dark Indulgence|5|10.0|buy|chocolate
4 2024-10-22 10:30:43|Raspberry Noir Temptation|8|8.0|buy|chocolate
5 2024-10-22 10:30:44|Velvet Cocoa Delight|3|5.0|buy|chocolate
6 2024-10-22 10:30:46|Golden Hazelnut Bliss|4|7.0|buy|chocolate
7 |
```

figure3: exemple de log généré

- Ajouter un Event Listener sur le Bouton "Buy"

Dans le système, chaque fois qu'un utilisateur clique sur le bouton "Buy" pour acheter un produit, un événement est déclenché. Pour capter cet événement et générer un fichier log, on a utilisé un event listener qui écoute cette action. On a fait cela à l'aide de Node.js, pour gérer cet événement. On crée une fonction qui sera appelée lorsque l'utilisateur clique sur le bouton "Buy". Ensuite, nous générons un fichier log avec les informations nécessaires.

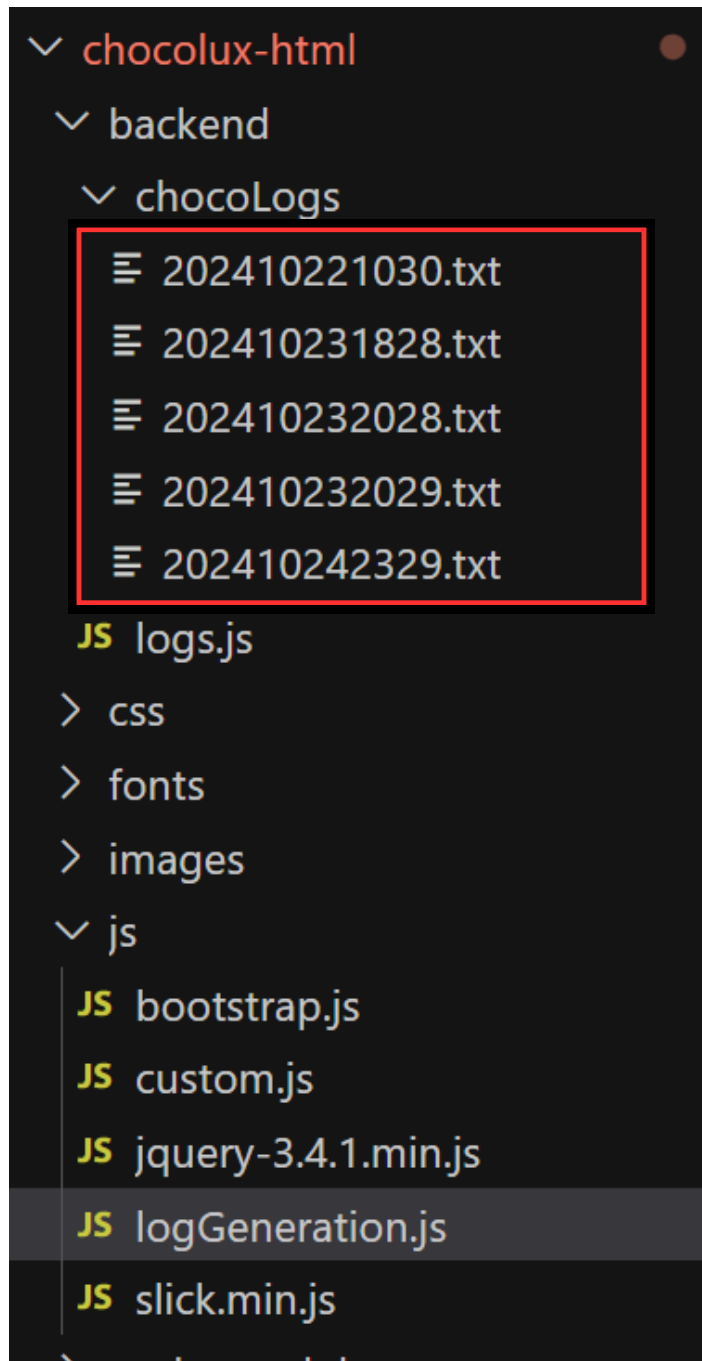


figure4: les logs générés

Bref ,Dans cette étape, nous avons ajouté un event listener au bouton "Buy" de chaque produit affiché sur le site. Ce listener capture chaque clic utilisateur et déclenche une fonction qui collecte les informations du produit (nom, identifiant, prix), ainsi que l'action effectuée et la page source. Cette méthode permet d'enregistrer précisément chaque interaction utilisateur pour un suivi détaillé des événements d'achat.

5.2 Collecte et Préparation des Données avec Apache Nifi

• Processus de Collecte des Logs

Pour la partie de collecte et préparation des données avec Apache NiFi, on a structuré et traité les fichiers générés par Node.js dans le dossier chocoLogs comme ceci :

1. Collecte des logs : Les fichiers de logs, générés avec un format YYYYMMDDHHMM.txt, sont placés dans le dossier chocoLogs. Ces fichiers contiennent les interactions des utilisateurs enregistrées par l'application Node.js, comme il montre la figure 2
2. Préparation avec Apache NiFi :
 - Configuration de NiFi : Nous avons configuré un flux NiFi pour traiter automatiquement les fichiers de logs. Le flux commence par une étape ListFile, qui surveille le dossier chocoLogs pour détecter les nouveaux fichiers ajoutés.
 - Structuration des fichiers : NiFi regroupe les fichiers en fonction du format YYYYMMDDHH. Par exemple, tous les logs générés à la même heure (indiquée par le préfixe YYYYMMDDHH) sont regroupés et déplacés dans un sous-dossier correspondant, tel que nifi/YYYYMMDDHH/.
 - Sortie structurée : Les fichiers sont organisés dans des sous-dossiers pour faciliter leur traitement ultérieur.

Dans cette étape, nous avons conçu un flux avec Apache NiFi pour automatiser la gestion des fichiers journaux générés par notre application Node.js. À l'aide des processeurs ListFile, FetchFile, UpdateAttribute et PutFile, le flux détecte les nouveaux fichiers dans un répertoire source, extrait leur contenu, modifie ou enrichit leurs attributs, puis les déplace dans un répertoire cible organisé par date (YYYYMMDDHH). Cette structuration facilite l'analyse et la gestion des données tout en garantissant un traitement efficace et automatisé.

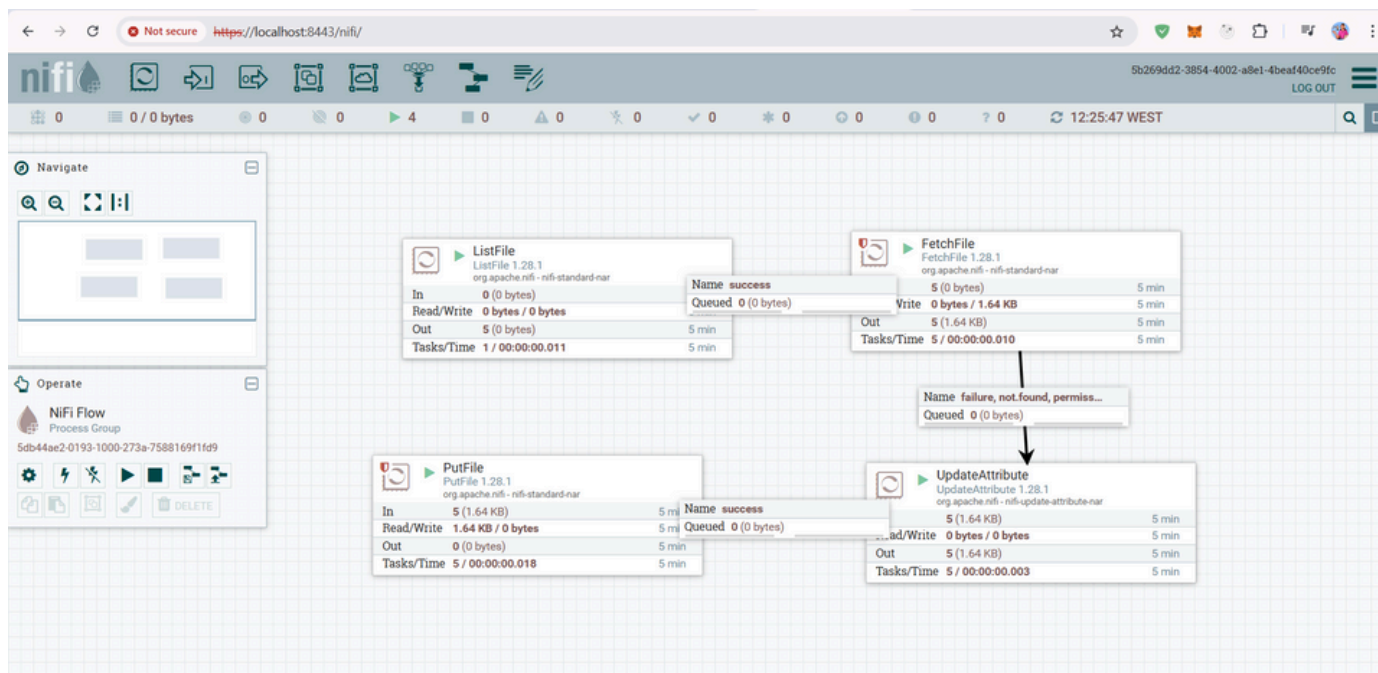


figure5: flux de traitement en apache Nifi

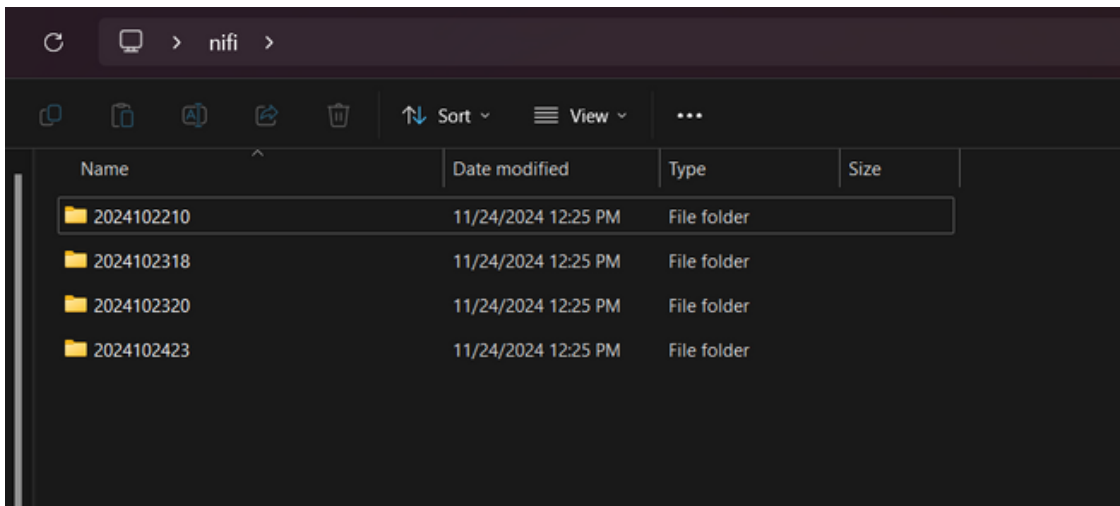


figure6: Les Répertoires structurés générés par Apache NiFi selon le format YYYYMMDDHH i

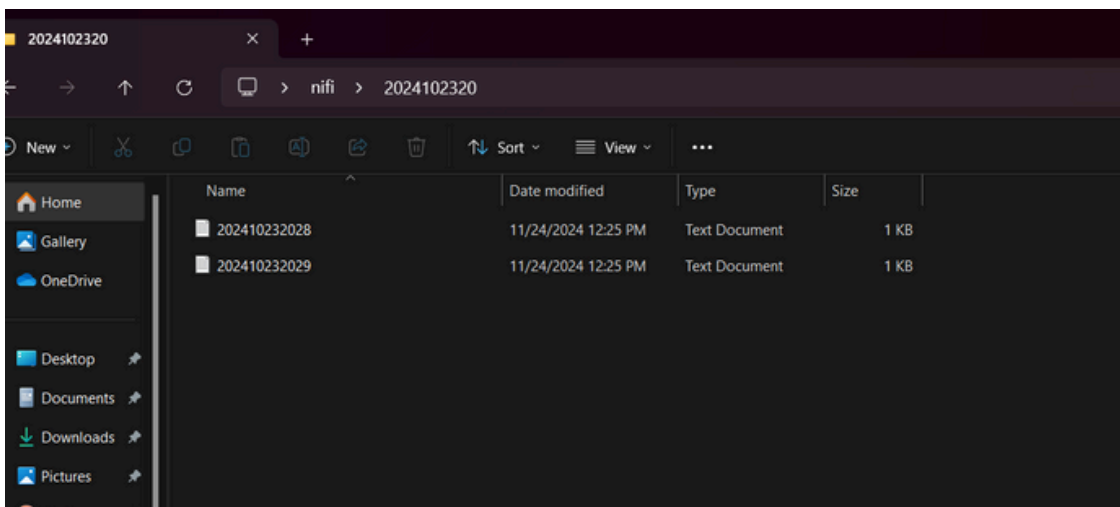


figure7: exemple des fichiers dans l'un de ces répertoirees

5.3 Traitement des Données avec Apache Spark

• Objectifs du Traitement des Données

L'objectif était de traiter les logs, d'extraire les informations pertinentes (comme le prix des ventes et le produit vendu) et d'agréger les données par période horaire. La sortie du traitement contient des fichiers qui comporte le nom sous formats m Date et heure : La date et l'heure de l'événement (format YMDh). ils contiennent les informations suivantes :

- Date : La date d'achat de chocolat (format YMD)
- Id : id de chocolat acheté.
- Produit : Le nom du produit.
- Somme des ventes : La somme des prix de toutes les ventes de ce produit durant l'heure spécifiée.

L'agrégation permet de réduire la complexité des données, en ne conservant que les informations essentielles pour la prise de décision.

Name	Date modified	Type	Size
2024102210	11/24/2024 12:50 PM	Text Document	1 KB
2024102318	11/24/2024 12:50 PM	Text Document	1 KB
2024102320	11/24/2024 12:50 PM	Text Document	1 KB
2024102423	11/24/2024 12:50 PM	Text Document	1 KB

figure8: les fichiers de sortie après l'exécution de script spark

```

2024/10/22 8|Raspberry Noir Temptation|8
2024/10/22 4|Golden Hazelnut Bliss|14
2024/10/22 5|Mystic Dark Indulgence|10
2024/10/22 3|Velvet Cocoa Delight|10

```

figure8: exemple de contenu d'un fichier parmi les fichiers de sortie après l'exécution de script spark

5.4 Création du Dashboard avec Python

Une fois les données agrégées dans le répertoire `./output/`, un script Python a été développé pour afficher un tableau de bord interactif. L'interface permet à l'utilisateur de spécifier une période (date de début et date de fin) pour visualiser les données correspondantes.

Trois champs permettent de filtrer les données affichées :

- Date de début : Permet de saisir une date de début au format YYYYMMDD.
- Date de fin : Permet de saisir une date de fin au format YYYYMMDD.
- Filtrer par article : Un champ texte optionnel pour affiner la recherche selon un nom ou un type d'article.

← → localhost:8501

Deploy

Filtres

Date de début (YYYYMMDD)

20241101

Date de fin (YYYYMMDD)

20241130

Filtrer par article

Charger les données

Dashboard des Ventes

figure9: dashboard en streamlit qui visualise l'analyse de notre données

Ce tableau de bord présente les métriques suivantes :

- Les produits les plus vendus pendant la période spécifiée.
- La somme des ventes pour chaque produit par période (heure, jour, etc.).

Des graphiques de tendance pour mieux visualiser les données.

Le tableau de bord est interactif et permet de filtrer les informations en fonction des dates,et par article offrant ainsi une vue d'ensemble et des insights utiles pour la prise de décision.

Donc si par exemple on entre des dattes pour visulaiser on obtinient le tableau suivant :

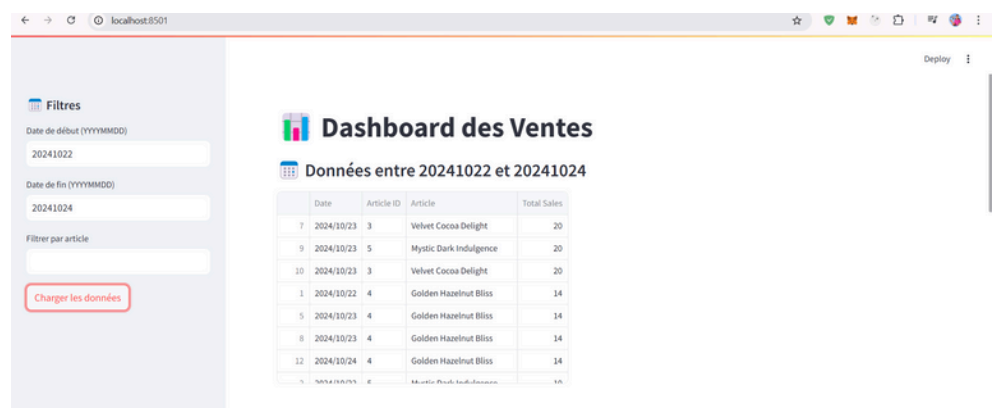


figure10: le résultat du dashboard en donnant une période spécifique

	Date	Article ID	Article	Total Sales
7	2024/10/23	3	Velvet Cocoa Delight	20
9	2024/10/23	5	Mystic Dark Indulgence	20
10	2024/10/23	3	Velvet Cocoa Delight	20
1	2024/10/22	4	Golden Hazelnut Bliss	14
5	2024/10/23	4	Golden Hazelnut Bliss	14
8	2024/10/23	4	Golden Hazelnut Bliss	14
12	2024/10/24	4	Golden Hazelnut Bliss	14
2	2024/10/22	5	Mystic Dark Indulgence	10
3	2024/10/22	3	Velvet Cocoa Delight	10
6	2024/10/23	5	Mystic Dark Indulgence	10
13	2024/10/24	3	Velvet Cocoa Delight	10
0	2024/10/22	8	Raspberry Noir Temptation	8
4	2024/10/23	8	Raspberry Noir Temptation	8
11	2024/10/24	8	Raspberry Noir Temptation	8

figure11: le tableau des ventes de chocolat dans cette période donnée et détaillé par heure et jour

Le tableau présenté dans le dashboard résume les données des ventes sur une période spécifique. Il affiche plusieurs colonnes clés : la date de chaque vente, l'identifiant de l'article (Article ID), le nom de l'article vendu, et le nombre total de ventes (Total Sales) pour chaque produit ("chocolat") à une heure donnée. Cela permet de visualiser rapidement les performances des différents articles, d'identifier les produits les plus vendus, et de détecter les tendances sur la période sélectionnée. Ce tableau constitue un outil essentiel pour analyser les résultats commerciaux de manière synthétique et organisée.

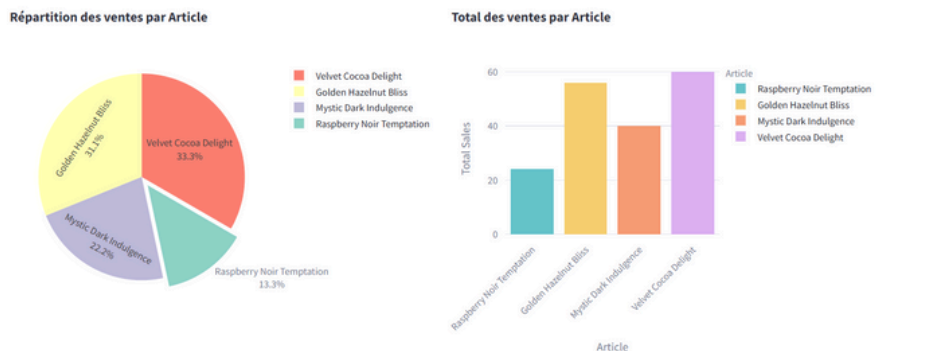


figure12: les graphes visualisant les ventes de chocolats

Ce visuel présente une analyse graphique des ventes par article durant une période spécifiée sous deux formats complémentaires.

- À gauche : Un diagramme circulaire (camembert) illustre la répartition des ventes par article en pourcentage. Il permet de visualiser facilement la part relative de chaque produit dans les ventes totales. Par exemple, "Velvet Cocoa Delight" constitue 33,3% des ventes, suivi de "Golden Hazelnut Bliss" (31,1%), reflétant les produits les plus populaires.
- À droite : Un diagramme en barres montre le total des ventes par article en valeurs absolues. Chaque barre correspond à un produit, permettant de comparer directement les performances des articles. On observe que "Velvet Cocoa Delight" et "Golden Hazelnut Bliss" dominent avec des ventes proches de 60 unités, tandis que "Raspberry Noir Temptation" est moins vendu.

Ces graphiques offrent une vue synthétique et comparative des performances des différents produits, facilitant l'identification des articles les plus et les moins performants.



figure12: les graphes visualisant les ventes de chocolats

• Graphique : Évolution des ventes par jour (Cumulé)

Le graphique linéaire présenté illustre l'évolution des ventes cumulées par jour sur la période sélectionnée. La courbe montre une progression constante des ventes, passant de 50 unités le 22 octobre 2024 à un total cumulé de 180 unités le 24 octobre 2024. Cette représentation permet de visualiser la croissance des ventes au fil du temps et d'identifier les jours où les performances ont été les plus significatives, comme entre le 22 et le 23 octobre, où une forte augmentation est observée.

• Statistiques : Total des ventes et article le plus vendu

1. La section "Statistiques" résume les informations clés des ventes :

Total des ventes : 180 unités, représentant l'ensemble des articles vendus sur la période sélectionnée.

2. Article le plus vendu : "Velvet Cocoa Delight", qui se distingue comme le produit ayant généré le plus grand nombre de ventes.

Ces statistiques fournissent une vue d'ensemble rapide et concise des performances globales et mettent en avant le produit phare.

on peut aussi filtré les données par article afin de se focaliser sur un article comme indiquent les figures suivants :

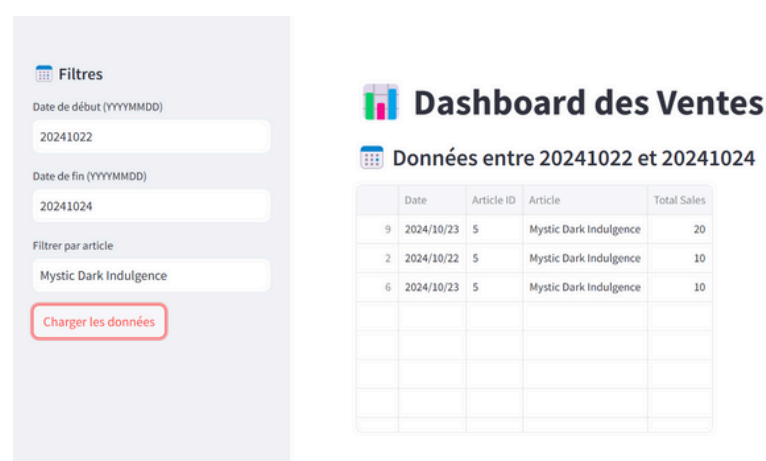


figure13: les graphes visualisant les ventes d'un catégorie de chocolats

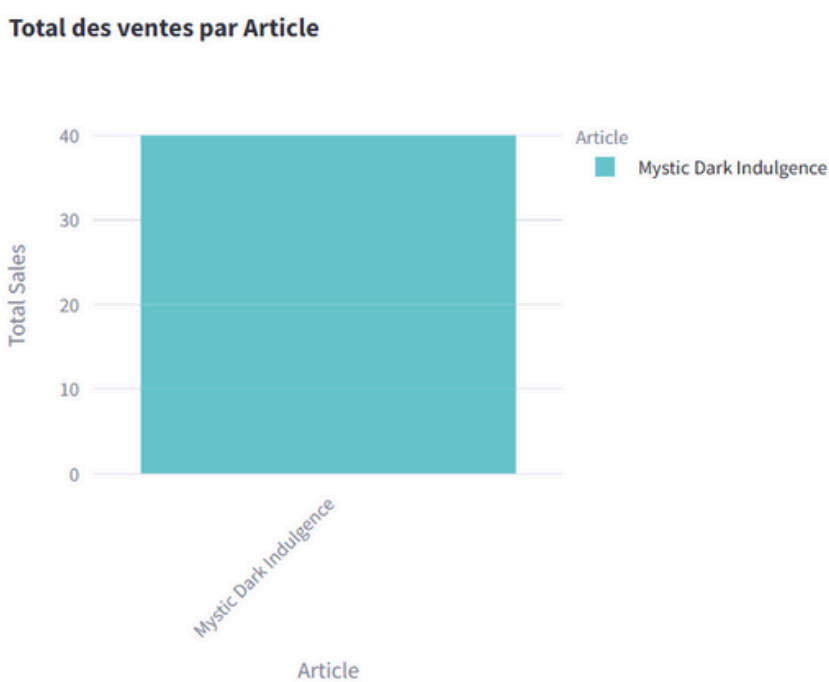


figure13: les graphes visualisant les ventes d'un catégorie de chocolats

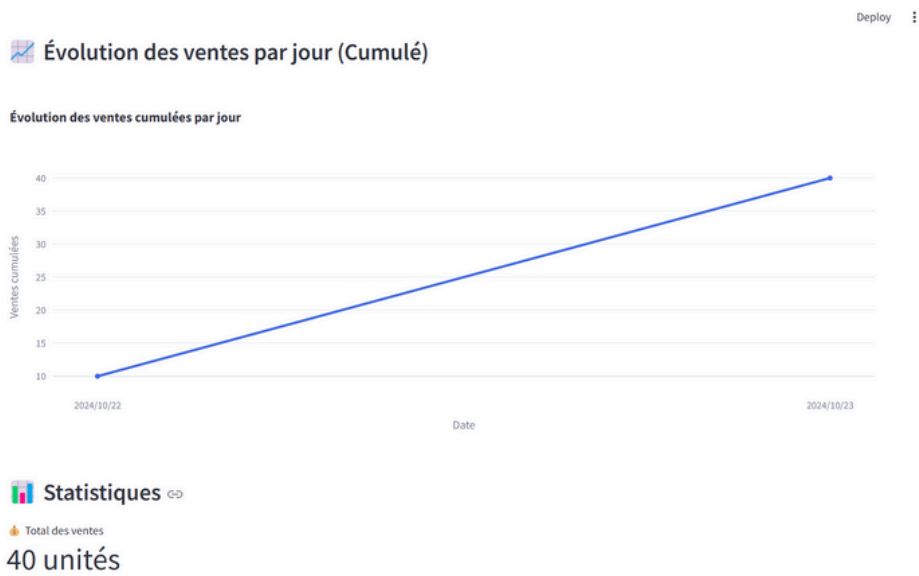


figure13: les graphes visualisant les ventes d'un catégorie de chocolats

Conclusion

Ce projet a permis d'acquérir une compréhension approfondie des outils utilisés par les Data Engineers, en particulier Apache Nifi, Apache Spark et Python pour la gestion de flux de données, le traitement à grande échelle, et la visualisation des résultats. Il met en lumière l'importance de l'automatisation dans le traitement des données et la création de tableaux de bord pour la prise de décision.

Perspectives

- Amélioration du Dashboard : Ajouter des fonctionnalités avancées, comme l'exportation des résultats sous différents formats (CSV, PDF, etc.).
- Intégration avec d'autres systèmes : Connecter le tableau de bord à une base de données pour permettre des requêtes en temps réel.
- Optimisation du traitement Spark : Afin de gérer plus efficacement les volumes de données croissants.

plus de détaille sur la partie code, visiter le lien github
[:https://github.com/elkadirizouhra/bigData](https://github.com/elkadirizouhra/bigData)