Université de Nouakchott

Faculté des Sciences et Techniques

Département de Mathématiques et Informatiques

Rapport de Projet

Analyse de Données massives et complexes

Collecte, stockage et analyse de données cinématographiques

Réalisé par :
Boudah Mohamed Lemine Ahmedou El Mokhtar
C20121

Encadré par :
Dr. Mohamed El Ghaly Beheitt

Année universitaire : 2025 – 2026

Projet réalisé dans le cadre du Master SSD – Statistiques et Sciences des Données

Table des matières

Introduction Générale

Le présent projet s'inscrit dans le cadre de notre formation universitaire, dédié à l'étude des technologies Big Data et des bases de données distribuées. Son objectif principal est de mettre en application les notions théoriques abordées en cours à travers la réalisation d'un projet pratique, structuré et représentatif d'un cas réel.

Face à l'augmentation continue du volume des données numériques, la capacité à collecter, stocker et analyser efficacement l'information est devenue une compétence essentielle dans le domaine de l'informatique et de la science des données. Ce projet vise ainsi à renforcer la maîtrise des différentes étapes du cycle de vie des données, depuis leur acquisition jusqu'à leur exploitation analytique.

Pour ce faire, nous avons choisi d'exploiter des données issues de la plateforme TMDB, spécialisée dans le domaine cinématographique. Ce choix repose sur la richesse et la diversité des informations disponibles, permettant d'appliquer des méthodes de traitement et d'analyse sur un ensemble de données réaliste et structuré.

Le travail réalisé couvre plusieurs étapes clés : la collecte automatisée des données par web scraping, leur stockage dans une base de données MongoDB reposant sur les mécanismes de sharding et de réplication, ainsi que le nettoyage, l'analyse et la visualisation des données à l'aide d'outils adaptés.

Ce rapport est organisé de manière progressive et cohérente. Il commence par la présentation des outils et de l'environnement de travail, puis décrit les données collectées et leur organisation. Il se termine par une phase d'analyse et d'interprétation des résultats obtenus, mettant en évidence les enseignements tirés de ce projet.

Chapitre 1

Outils et environnement de travail

1.1 Environnement matériel et système

Le présent projet a été réalisé sur un ordinateur personnel, sans recours à un cluster physique dédié. La machine utilisée dispose de 8 Go de mémoire RAM, ce qui a nécessité une configuration maîtrisée et optimisée des services Big Data, notamment lors de la mise en place du sharding et de la réplication MongoDB.

Le système d'exploitation utilisé est Windows 10 (64 bits), choisi pour sa stabilité et sa compatibilité avec les outils de développement employés dans le cadre de ce travail.

1.2 Langages et bibliothèques utilisées

Le langage principal utilisé dans ce projet est Python, en raison de sa simplicité, de sa richesse en bibliothèques et de son adoption massive dans les domaines du Big Data et de l'analyse de données.

Plusieurs bibliothèques Python ont été mobilisées, notamment :

— **requests** pour l'envoi de requêtes HTTP lors du web scraping ;
— **BeautifulSoup** pour l'analyse et l'extraction des données HTML ;
— **pandas** pour la manipulation, le nettoyage et l'analyse des données ;
— **matplotlib** pour la génération des visualisations statistiques ;

— **pymongo** pour l'interaction entre Python et la base de données MongoDB.

1.3 Outils de développement

L'environnement de développement intégré (IDE) utilisé est PyCharm, qui offre des fonctionnalités avancées telles que la gestion des environnements virtuels, le débogage interactif et l'intégration native avec Python.

Pour l'analyse exploratoire et la visualisation des données, l'outil Jupyter Notebook a également été utilisé, permettant une approche itérative et interactive de l'analyse statistique.

1.4 Technologies Big Data utilisées

La technologie Big Data centrale de ce projet est MongoDB, une base de données NoSQL orientée documents, particulièrement adaptée au stockage de données semi-structurées au format JSON.

Une architecture distribuée a été mise en place, combinant :

— le **sharding**, afin de répartir les données sur plusieurs nœuds ;

— la **réplication** (Replica Sets), afin d'assurer la tolérance aux pannes et la haute disponibilité.

Cette configuration permet de simuler, à l'échelle d'un ordinateur personnel, une architecture Big Data proche de celles utilisées en environnement professionnel.

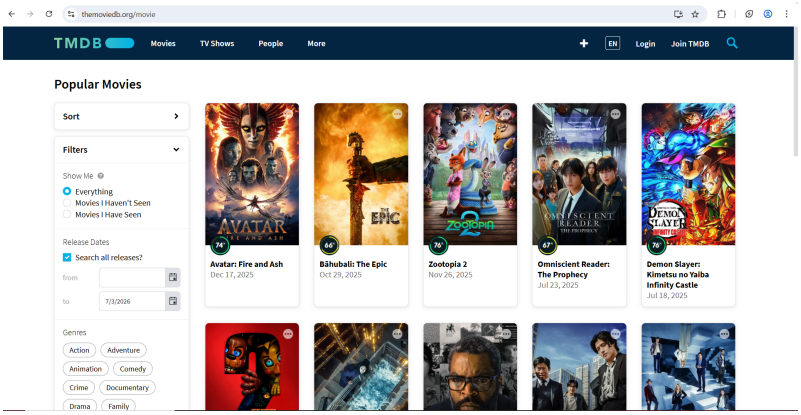
Chapitre 2

Présentation des données

2.1 Source des données

Les données utilisées dans ce projet proviennent du site **The Movie Database (TMDB)**, une base de données en ligne collaborative dédiée aux films, séries télévisées et contenus audiovisuels.

TMDB offre des informations riches et régulièrement mises à jour concernant les œuvres cinématographiques, ce qui en fait une source pertinente pour des analyses statistiques liées à l'industrie du cinéma.



2.2 Nature et format des données

Les données collectées sont de nature semi-structurée et sont stockées au format **JSON**. Ce format est particulièrement adapté aux bases de données NoSQL, car il permet une représentation flexible et hiérarchique de l'information.

Chaque enregistrement correspond à un film unique et regroupe plusieurs attributs décrivant ses caractéristiques principales.

2.3 Variables collectées

Dans le cadre de ce projet, les variables suivantes ont été extraites pour chaque film :

— **title** : nom du film ;
— **release-date** : date complète de sortie officielle ;
— **year** : Année de sortie, extraite à partir de la date de sortie ;
— **rating** : évaluation attribuée par les utilisateurs.

Ces variables ont été sélectionnées de manière à permettre une analyse temporelle et comparative des évaluations des films.

2.4 Volume et structure du jeu de données

Le jeu de données final est composé de **1 000 films**, conformément aux exigences du projet. Chaque film est représenté par un document JSON indépendant.

Cette volumétrie permet de réaliser des analyses statistiques pertinentes tout en restant compatible avec les ressources limitées d'un ordinateur personnel.

Chapitre 3

Collecte des données (Web Scraping)

3.1 Principe du Web Scraping

Le Web Scraping désigne un ensemble de techniques permettant d'extraire automatiquement des informations à partir de pages web. Cette approche repose généralement sur l'envoi de requêtes HTTP et l'analyse du code HTML des pages ciblées.

Dans le cadre de ce projet, le Web Scraping a été utilisé afin de constituer un jeu de données original, répondant aux contraintes imposées en termes de volume et de structure.

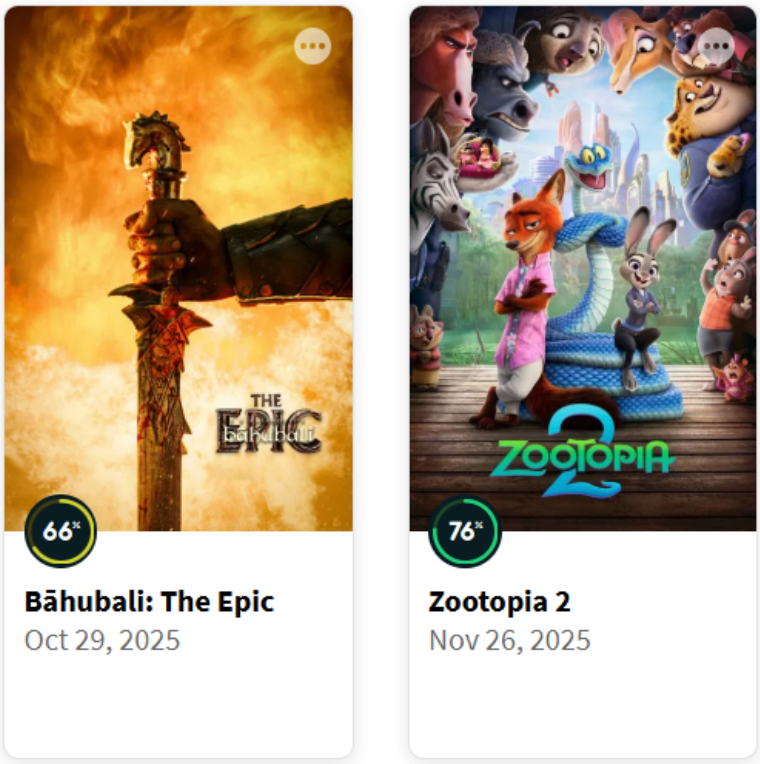
3.2 Choix du site TMDB

Le site **The Movie Database (TMDB)** a été retenu comme source de données en raison de la richesse de son contenu, de la clarté de la structure de ses pages et de la cohérence des informations proposées.

De plus, TMDB permet un accès public à de nombreuses pages sans nécessiter d'authentification, ce qui simplifie le processus de collecte automatisée.

3.3 Structure des pages web

Les pages du site TMDB présentent une structure HTML relativement homogène, chaque film étant représenté par un bloc contenant les informations essentielles telles que le titre, la date de sortie et la note moyenne.



Cette homogénéité facilite l'identification des balises HTML pertinentes et permet une extraction systématique des données souhaitées.

3.4 Méthodologie de collecte

La collecte des données s'est déroulée en plusieurs étapes. Tout d'abord, des requêtes HTTP ont été envoyées vers les pages de films à l'aide de la bibliothèque `requests`. Ensuite, le contenu HTML récupéré a été analysé à l'aide de la bibliothèque `BeautifulSoup` afin d'extraire les informations ciblées.

Un mécanisme de parcours automatique des pages a été mis en place, permettant de collecter progressivement un nombre suffisant d'enregistrements.

3.5 Stockage des données collectées

Les données extraites ont été stockées dans un fichier au format **JSON**, chaque film étant représenté par un objet indépendant.

Ce choix de format facilite à la fois l'importation ultérieure dans MongoDB et la manipulation des données lors des phases de nettoyage et d'analyse.

Chapitre 4

Stockage distribué des données avec MongoDB

4.1 Introduction sur MongoDB

MongoDB est un système de gestion de bases de données orienté documents, largement utilisé dans les environnements Big Data. Il permet de stocker des données semi-structurées sous forme de documents JSON, offrant ainsi une grande flexibilité dans la modélisation des données.

Dans ce projet, MongoDB a été utilisé comme solution de stockage distribué afin de gérer efficacement le volume de données collectées par web scraping et de mettre en pratique les concepts de sharding et de réplication.

4.2 Architecture distribuée du cluster

L'architecture mise en place repose sur un cluster MongoDB distribué, conçu et configuré localement sur une seule machine. Le cluster est composé de six serveurs distincts, chacun exécuté dans une fenêtre de commande indépendante (*CMD*).

Cette architecture comprend :

— deux shards (*Shard 1* et *Shard 2*) ;
— chaque shard est configuré sous forme de *replica set* composé d'un nœud primaire et d'un nœud secondaire ;
— un serveur de configuration (*Config Server*) ;
— un routeur MongoDB (`mongos`).

L'ensemble des services fonctionne sur `localhost`, en utilisant des ports compris entre `27017` et `27022`. Le routeur `mongos` joue un rôle central en recevant les requêtes clientes et en les redirigeant automatiquement vers les shards appropriés.

4.3 Configuration du cluster MongoDB

La configuration du cluster constitue l'étape centrale de ce projet. Elle a été réalisée manuellement afin de mieux comprendre le fonctionnement interne de MongoDB dans un environnement distribué.

4.3.1 Organisation des répertoires de données

Un répertoire principal nommé `tmdb_data` a été créé sur le disque `C:`. Il contient les sous-dossiers suivants :

— `config` ;
— `shard1_primary` ;
— `shard1_secondary` ;
— `shard2_primary` ;
— `shard2_secondary`.

Chaque dossier est associé à un nœud MongoDB spécifique et contient les fichiers de données correspondants.

4.3.2 Mise en place des shards et de la réplication

Chaque shard a été configuré sous forme de replica set afin d'assurer une tolérance aux pannes et une meilleure disponibilité des données. Les nœuds primaires assurent les opérations d'écriture, tandis que les nœuds secondaires maintiennent des copies synchronisées des données.

Le sharding a ensuite été activé afin de permettre la distribution automatique des données entre les deux shards du cluster.

À titre illustratif, l'exemple suivant présente la commande utilisée pour le lancement d'un nœud primaire d'un shard configuré en replica set.

```
mongod --shardsvr --replSet shard1Repl --port 27018
--dbpath C:\tmdb_data\shard1_primary --bind_ip localhost
```

Les autres nœuds ont été configurés de manière similaire, en adaptant le port, le rôle et le répertoire de données.

4.3.3 Serveur de configuration et routeur mongos

Le serveur de configuration (*Config Server*) est chargé de stocker les métadonnées du cluster, notamment la répartition des données entre les shards et la configuration du sharding. Il ne contient aucune donnée applicative.

Le routeur `mongos` constitue le point d'entrée unique du cluster. Toutes les opérations clientes transitent par ce service, qui se charge de rediriger les requêtes vers les shards appropriés de manière transparente.

Le serveur de configuration a été lancé en mode `configsvr`, comme illustré par la commande suivante :

```
mongod --configsvr --replSet configRepl --port 27019
--dbpath C:\tmdb_data\config --bind_ip localhost
```

Le routeur `mongos` a été configuré pour se connecter au serveur de configuration et assurer le routage des requêtes :

```
mongos --configdb configRepl/localhost:27019 --port 27017
--bind_ip localhost
```

4.4 Importation des données

Les données collectées par web scraping ont été stockées dans un fichier JSON structuré contenant environ 1 000 documents. L'importation a été réalisée à l'aide de l'outil `mongoimport`, fourni nativement par MongoDB.

L'importation a été effectuée via le routeur `mongos`, garantissant ainsi une répartition automatique des données entre les shards du cluster.

```
mongoimport --host localhost --port 27017 \
--db tmdbDB --collection movies \
--file C:\tmdb_data\tmdb_movies.json --jsonArray
```

Cette approche permet une intégration efficace des données dans un environnement distribué sans intervention manuelle sur la distribution.

4.5 Automatisation du démarrage du cluster

Afin de simplifier l'exploitation du cluster MongoDB, des scripts `.bat` ont été développés pour automatiser le démarrage des différents services.

Les scripts suivants ont été créés :

- `start_config.bat` ;
- `start_shard1.bat` ;
- `start_shard2.bat` ;
- `start_mongos.bat` ;
- `start_all.bat`.

Ces fichiers sont regroupés dans le répertoire `C:\tmdb_cluster` et permettent de lancer l'ensemble du cluster en une seule commande.

Cette automatisation garantit un démarrage rapide, cohérent et reproductible du cluster après chaque redémarrage du système.

Chapitre 5

Analyse statistique et visualisations

Ce chapitre est consacré à l'analyse et à la visualisation des données collectées depuis la base de données MongoDB. L'objectif est de préparer les données, de vérifier leur cohérence, puis d'extraire des informations pertinentes à travers des analyses statistiques et des représentations graphiques.

L'ensemble des traitements a été réalisé dans un environnement Jupyter Notebook en utilisant le langage Python et des bibliothèques spécialisées en science des données.

5.1 Préparation et nettoyage des données

Extraction et structuration des données

Les données ont été extraites depuis la collection `movies` de la base de données `tmdbDB` en utilisant la bibliothèque `PyMongo`. Les documents récupérés ont été stockés dans une liste Python, puis convertis en un DataFrame afin de faciliter les opérations de traitement et d'analyse.

Exploration initiale du jeu de données

Une première exploration des données a été réalisée afin d'examiner la structure générale du jeu de données. Cette étape a permis de vérifier la présence éventuelle de valeurs manquantes ainsi que la cohérence globale des informations. Les résultats obtenus ont confirmé l'absence de valeurs nulles dans les champs analysés.

Correction des types et normalisation des variables

Une attention particulière a été portée aux types des variables. Le champ `rating`, initialement de type textuel, a été converti en type numérique afin de permettre les calculs statistiques. Par ailleurs, les valeurs des évaluations

ont été normalisées en les divisant par 10, ce qui permet d'exprimer les notes sur une échelle de 0 à 10 au lieu de 0 à 100, rendant ainsi les analyses plus lisibles.

Vérification de la cohérence des données

Des contrôles supplémentaires ont été effectués afin de garantir la validité des données. Aucune valeur de note ne se situe en dehors de l'intervalle $[0, 10]$, et aucune année de sortie ne dépasse l'intervalle $[1900, 2025]$. Ces vérifications assurent la fiabilité des analyses ultérieures.

Traitement des films non évalués

L'exploration des données a révélé la présence de films ayant une note égale à zéro. Ces films correspondent à des œuvres anciennes ou non évaluées, dont la note ne reflète pas une appréciation réelle. Par conséquent, ces observations ont été supprimées du jeu de données afin de ne pas biaiser les résultats.

Cette opération a entraîné une réduction du nombre de films de 1000 à 986.

5.2 Statistiques descriptives des données

Présentation des statistiques descriptives

Avant de procéder aux visualisations, une analyse statistique descriptive a été réalisée afin de résumer les principales caractéristiques du jeu de données. Cette analyse inclut des indicateurs tels que la moyenne, la médiane, l'écart-type, ainsi que les valeurs minimales et maximales des variables numériques.

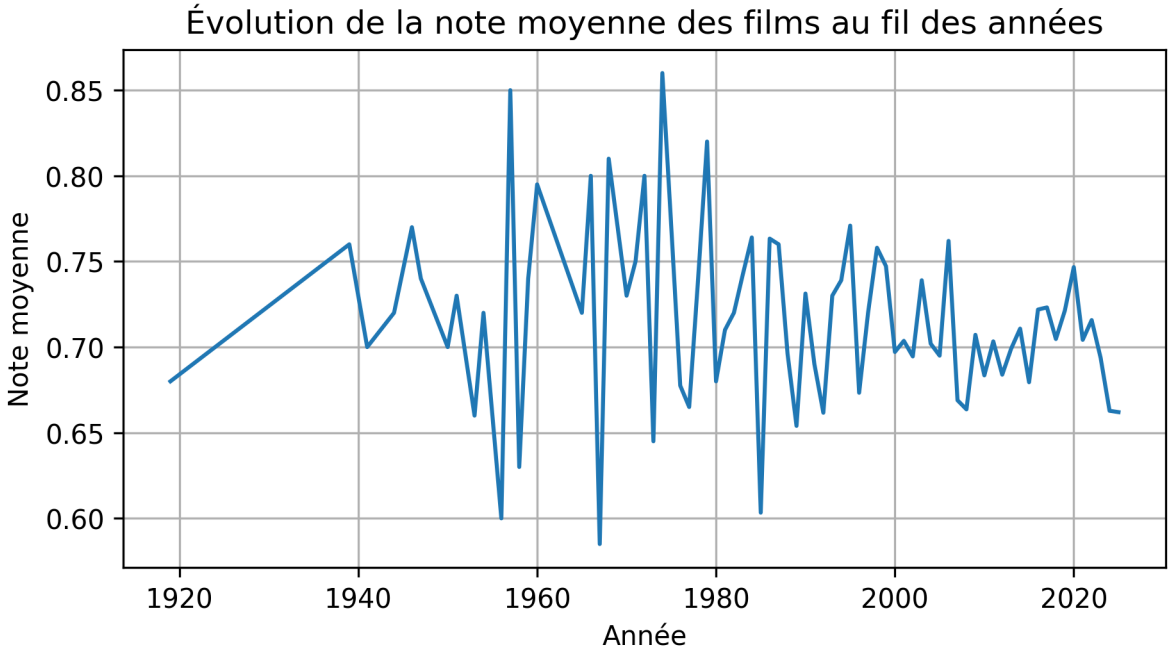
Table de statistiques descriptives

	count	mean	std	min	25%	50%	75%	max
year	986.000000	2013.006085	15.857294	1919.000000	2007.000000	2019.000000	2025.000000	2025.000000
rating	986.000000	0.690629	0.092816	0.200000	0.640000	0.700000	0.760000	0.900000

5.3 Visualisation des données

Après la phase de préparation et l'analyse descriptive, plusieurs visualisations ont été réalisées afin d'explorer les données sous différents angles. Chaque graphique est accompagné d'une interprétation permettant d'expliquer les résultats observés.

Évolution de la note moyenne des films au fil des années



Ce graphique représente l'évolution de la note moyenne des films en fonction de leur année de sortie.

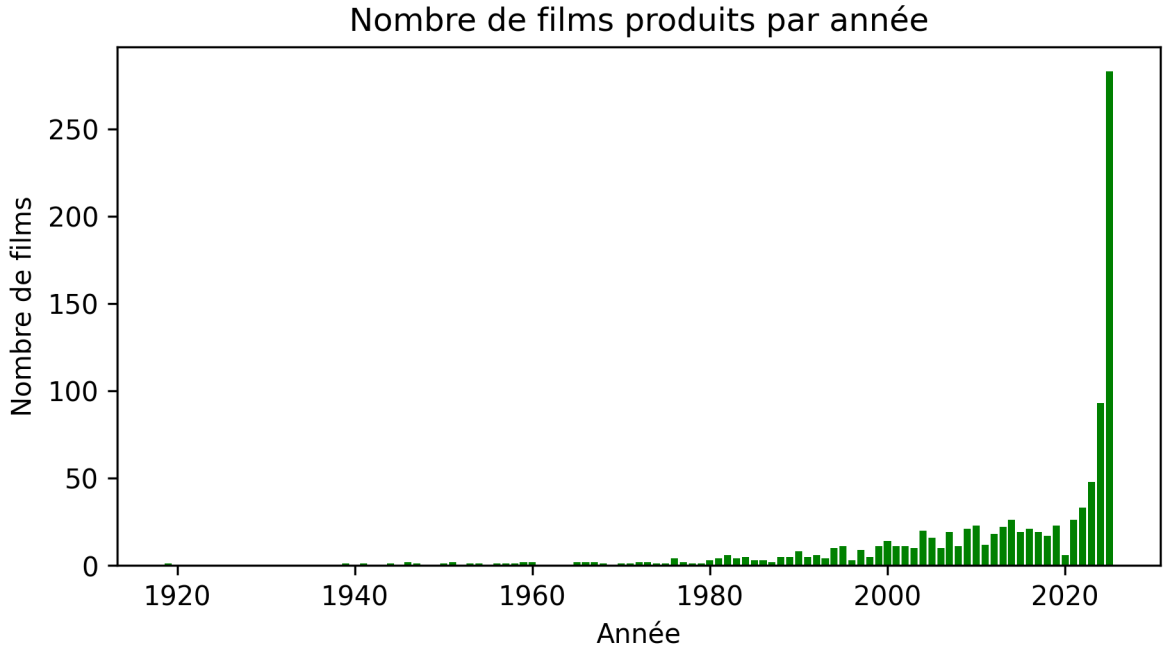
On observe que les notes moyennes présentent des fluctuations relativement importantes sur l'ensemble de la période étudiée, en particulier pour les années anciennes. Ces variations peuvent s'expliquer par le nombre limité de films produits durant certaines années, ce qui rend la moyenne plus sensible aux valeurs extrêmes.

À partir des années 1950 jusqu'aux années 1980, la note moyenne oscille globalement entre 6,5 et 8,0 , avec quelques pics notables dépassant 8,5 . Cette période semble marquée par une forte variabilité des évaluations.

À partir des années 1990, les notes moyennes deviennent plus stables, se concentrant majoritairement autour de l'intervalle [6,8 ; 7,5]. Cette stabilisation peut être liée à l'augmentation du nombre de films produits et évalués, ce qui tend à lisser les moyennes annuelles.

Globalement, aucune tendance strictement croissante ou décroissante ne se dégage sur le long terme, mais le graphique met en évidence une évolution plus régulière des notes dans les périodes récentes.

Nombre de films produits par année



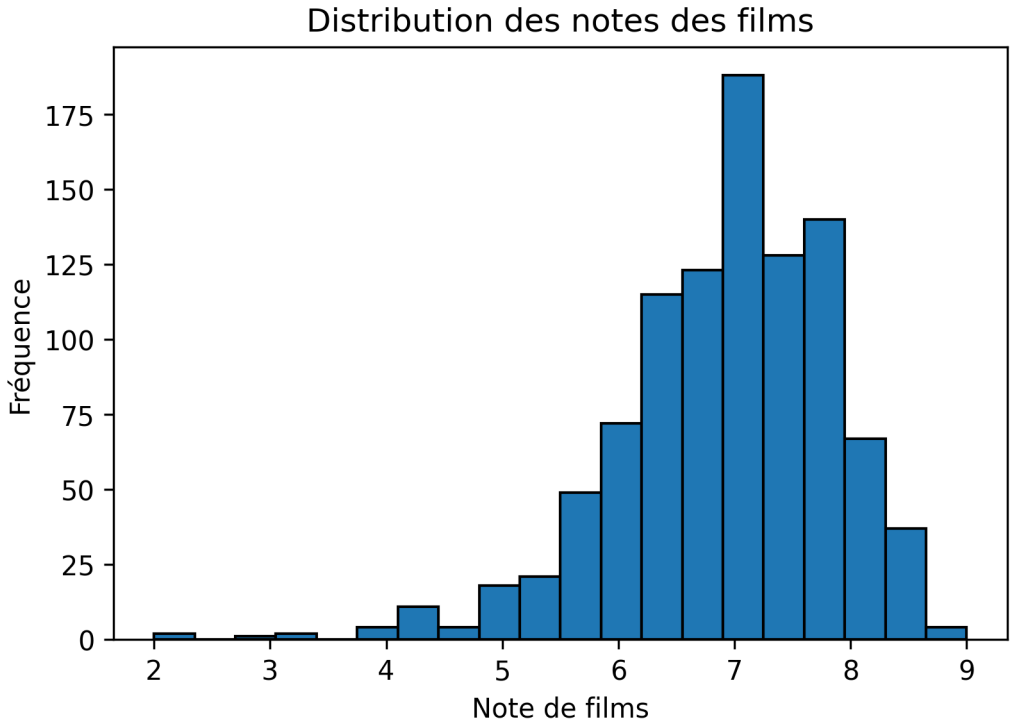
Cette visualisation met en évidence la variation du nombre de films produits au fil des années.

Jusqu'aux années 1970, la production cinématographique reste relativement faible et irrégulière, ce qui s'explique par les limites technologiques, économiques et industrielles de l'époque.

À partir des années 1980, on observe une augmentation progressive du nombre de films produits, qui s'accentue fortement après les années 2000. Cette croissance devient particulièrement significative au cours des années récentes, où le nombre de films atteint un niveau nettement supérieur aux périodes précédentes.

Cette évolution reflète l'industrialisation croissante du secteur cinématographique, la démocratisation des outils de production, ainsi que l'émergence des plateformes numériques, favorisant une production plus abondante et diversifiée.

Distribution des notes des films



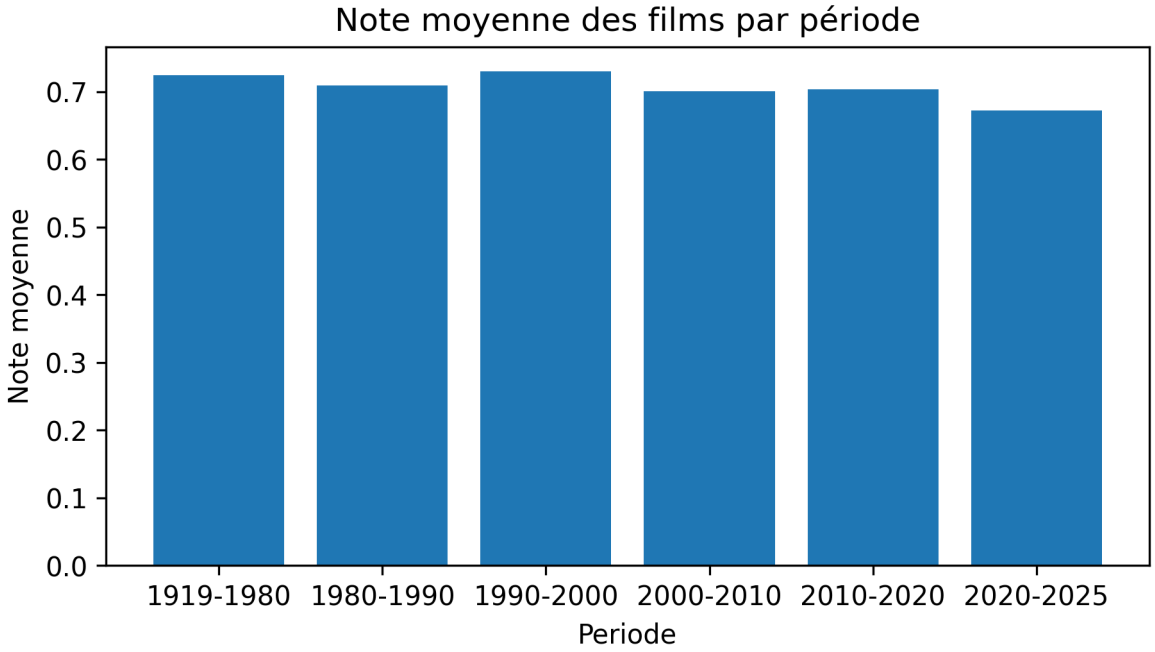
L'histogramme représente la distribution des notes des films après normalisation sur une échelle de 0 à 10.

On observe une forte concentration des notes dans l'intervalle compris entre 6 et 8, avec un pic autour de la note 7, ce qui indique que la majorité des films reçoivent des évaluations globalement positives.

Les notes très faibles (inférieures à 4) ainsi que les notes très élevées (supérieures à 8,5) sont relativement rares, ce qui suggère une distribution asymétrique centrée sur des évaluations moyennes à bonnes.

Cette répartition traduit une tendance générale des utilisateurs à attribuer des notes modérées, les jugements extrêmes étant moins fréquents.

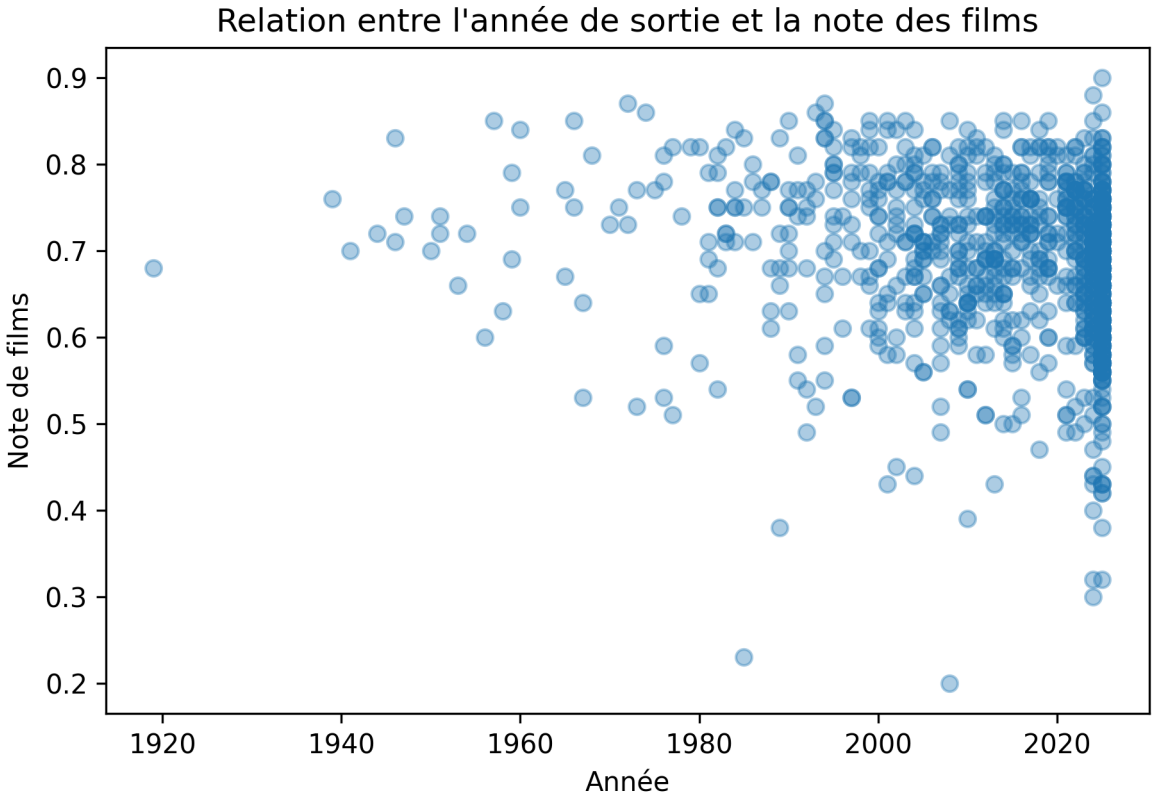
Note moyenne des films par période



Ce graphique compare la note moyenne des films selon différentes périodes temporelles.

Les valeurs sont légèrement plus élevées durant les périodes 1919–1980 et 1990–2000. À partir de 2000–2010, on observe une légère baisse progressive, qui se prolonge jusqu'à 2020–2025, pouvant s'expliquer par une évolution des critères d'évaluation, une diversification accrue de la production cinématographique ou un changement des attentes du public. Globalement, l'absence de fortes variations traduit une stabilité globale des évaluations dans le temps.

Relation entre l'année de sortie et la note des films



Cette représentation permet d'observer la relation entre l'année de sortie des films et leur évaluation.

Le nuage de points met en évidence une absence de relation linéaire forte entre l'année de sortie et la note des films. Les notes restent majoritairement concentrées entre 6.0 et 8.0 quelle que soit la période. On observe toutefois une dispersion plus importante pour les films récents, traduisant une plus grande hétérogénéité des évaluations, probablement liée à l'augmentation du volume de production et à la diversité des genres et des publics. Globalement, l'année de sortie n'apparaît pas comme un facteur déterminant de la note.

Conclusion du chapitre

Ce chapitre a permis de préparer, analyser et visualiser les données issues de la base MongoDB. Les résultats obtenus mettent en évidence l'intérêt de l'analyse des données pour extraire des informations pertinentes à partir de bases de données volumineuses.

Conclusion générale

Ce projet a été mené en respectant l'ensemble des étapes demandées, depuis la collecte des données jusqu'à leur analyse. Il a débuté par un web scraping du site TMDB en utilisant le langage Python, permettant l'extraction de 1000 films, stockés initialement dans un fichier JSON.

Les données ont ensuite été importées dans une base MongoDB distribuée, déployée sur un cluster composé de deux shards, chacun disposant d'une réplique afin d'assurer la disponibilité et la tolérance aux pannes. L'architecture du cluster a été automatisée à l'aide de scripts .bat, facilitant son démarrage et sa gestion, tandis que l'importation des données a été réalisée via mongoimport.

Dans une phase ultérieure, les données ont été chargées depuis MongoDB vers un Jupyter Notebook à l'aide de PyMongo, puis nettoyées, préparées et explorées. Cette préparation a permis de garantir la qualité des données avant l'analyse. Enfin, une analyse descriptive accompagnée de cinq visualisations a été réalisée afin de mettre en évidence les principales tendances liées à la production et aux notes des films.

En conclusion, ce projet illustre l'intérêt d'une approche complète combinant web scraping, bases de données NoSQL distribuées et analyse de données, et démontre la cohérence entre les choix techniques effectués et les objectifs analytiques visés.

Ce travail marque la fin de ce projet .
