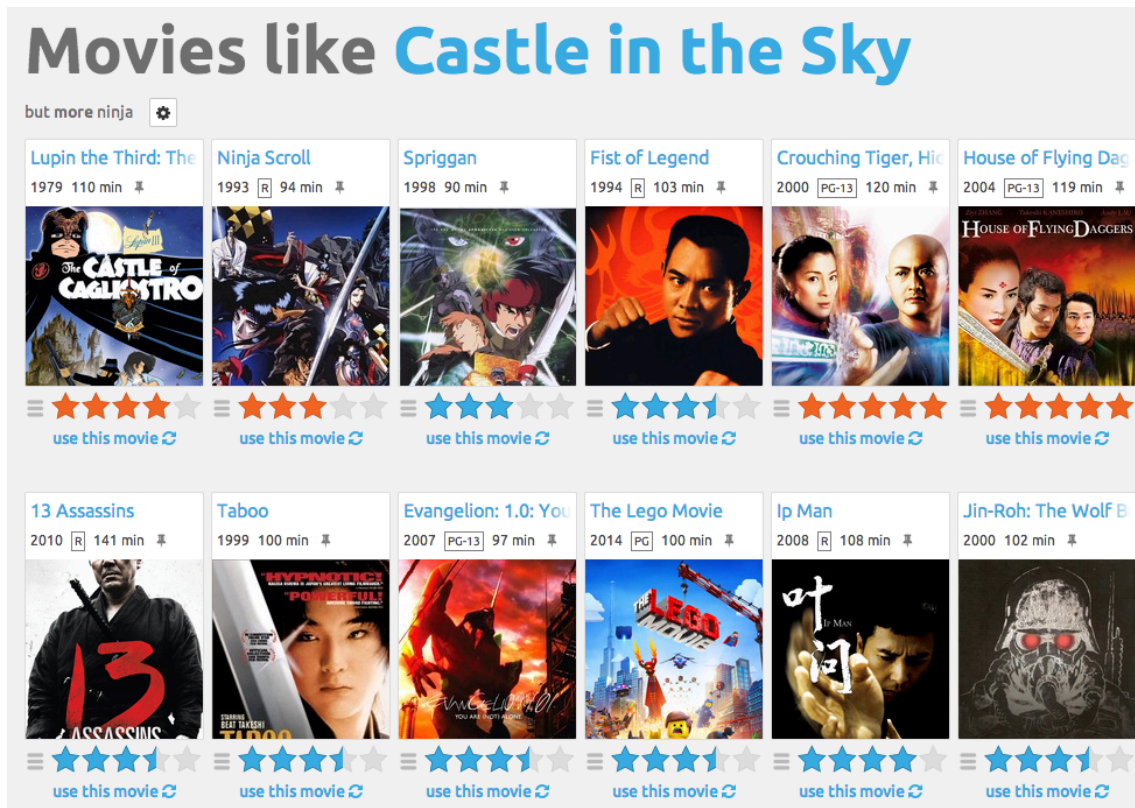


Sparkle Movie

Apache Spark peut traiter des pétaoctets de données en quelques secondes en distribuant les tâches sur plusieurs machines ! ⚡

MovieLens

Vous travaillez pour une plateforme de streaming vidéo qui souhaite améliorer l'expérience utilisateur en proposant des recommandations personnalisées. Vous devez utiliser l'ensemble de données MovieLens pour créer un modèle de recommandation et fournir une liste de films recommandés pour différents utilisateurs.



The screenshot shows the 'Movies like Castle in the Sky' recommendation page on the MovieLens website. The page features a grid of 12 movie cards, each with a title, year, rating, duration, and a star rating. Below each card is a 'use this movie' button with a circular arrow icon.

Movie Title	Year	Rating	Duration
Lupin the Third: The Castle of Cagliostro	1979	110 min	
Ninja Scroll	1993	R	94 min
Spriggan	1998		90 min
Fist of Legend	1994	R	103 min
Crouching Tiger, Hidden Dragon	2000	PG-13	120 min
House of Flying Daggers	2004	PG-13	119 min
13 Assassins	2010	R	141 min
Taboo	1999		100 min
Evangelion: 1.0: You Are (Not) Alone	2007	PG-13	97 min
The Lego Movie	2014	PG	100 min
Ip Man	2008	R	108 min
Jin-Roh: The Wolf Child	2000		102 min



Le dataset contient les informations suivantes :

1. **ratings.csv** : Les notes attribuées par les utilisateurs aux films.
 - userId, movieId, rating, timestamp
2. **movies.csv** : Les métadonnées des films.
 - movieId, title, genres

1. Préparation de l'environnement

- Installez PySpark sur votre machine.
- Configurez une session Spark.
- Récupérez le dataset MovieLens depuis [cette page](#), selon vos ressources, différentes tailles sont disponibles.

2. Chargement et exploration des données

- Importez les fichiers **ratings.csv** et **movies.csv** dans des DataFrames Spark.
- Affichez les 10 premières lignes de chaque DataFrame pour en comprendre la structure.
- Nettoyez les données si nécessaire (valeurs manquantes, doublons, etc.).
- Analysez les tendances générales :
 - Quels sont les films les mieux notés en moyenne ?
 - Quels genres de films sont les plus populaires ?
- Générez différentes visualisations à l'aide de bibliothèques Python ainsi que l'outil **Tableau Desktop**.

3. Modélisation avec Spark MLlib : ALS



- Utilisez l'algorithme **ALS (Alternating Least Squares matrix factorization)** de Spark MLlib pour entraîner un modèle de recommandation.
- Ajustez les hyperparamètres comme le **rank**, la **régularisation** (regParam) et le nombre d'itérations.
- Utilisez des métriques comme la **Root Mean Square Error (RMSE)** pour évaluer la performance du modèle.

4. Recommandation basée sur le contenu

- Créez des profils de films basés sur les genres.
- Implémentez un système recommandant des films similaires à ceux aimés par un utilisateur (TF-IDF, similarité cosinus).

5. Recommandation Basée sur les Proximités Utilisateurs (KNN)

- Implémentez une approche KNN pour trouver des utilisateurs similaires.
- Générez des recommandations en fonction des évaluations des voisins proches.

6. Evaluation des approches de recommandation

- Évaluez la précision et la couverture des recommandations.
- Comparez les résultats des différentes approches.
- Donnez des recommandations pour **3 à 5 utilisateurs fictifs**.
- Concluez sur la performance de vos différentes méthodes.

Compétences visées

→ Analyse de données



→ Apprentissage automatique

Rendu

L'évaluation de ce projet se fera sur deux aspects :

1. Une présentation explicative de votre travail sous forme de diapositives.
2. Un repository github public nommé **sparkle-movie**, contenant les éléments suivants :
 - a. Un **notebook Python propre et commenté** (introduction, titres des sections, interprétation des visuels, conclusion, etc) contenant le procédé de développement de votre outil, du nettoyage à la modélisation des données, en passant par **l'analyse exploratoire. Pensez à répondre à la problématique.** Vous pouvez avoir au maximum deux notebooks, un pour l'exploration et l'autre pour la modélisation de données.
 - b. Un fichier **README.md** présentant le contexte du projet, les données et leur analyse, les algorithmes utilisés et une conclusion sur votre travail.

Base de connaissances

- [PySpark Overview](#)
- [Qu'est-ce que Tableau Desktop ?](#)