

Mon système de recommandation de  
films



## Introduction

« [Netflix](#) est un service de diffusion en streaming qui permet à ses membres de regarder une grande variété de séries TV, films, documentaires, etc. sur des milliers d'appareils connectés à Internet. »

Créé en 1998, Netflix pèse aujourd'hui plus de 20 milliards de dollars de chiffre d'affaires et consomme 12,6% de la bande passante Internet mondiale.

Lorsqu'on accède au service Netflix, le système de recommandations aide l'utilisateur à trouver aussi facilement que possible les séries TV ou films qu'il pourrait apprécier, grâce à un système de recommandation. Netflix calcule ainsi la probabilité que l'utilisateur regarde un titre donné du catalogue de Netflix, et peut ainsi optimiser ces partenariats ou plus globalement sa stratégie marketing. Netflix est l'archétype de la société *data-driven*.

**Votre client n'est pas Netflix, mais il a de grandes ambitions !**

# Objectif

Vous êtes un Data Analyst freelance. Un cinéma en perte de vitesse situé dans la Creuse vous contacte. Il a décidé de passer le cap du digital en créant un site Internet taillé pour les locaux.



Pour aller encore plus loin, il vous demande de [créer un moteur de recommandations de films](#) qui à terme, enverra des notifications aux clients via Internet.

Pour l'instant, aucun client n'a renseigné ses préférences, **vous êtes dans une situation de cold start**. Mais heureusement, le client vous donne une base de données de films basée sur la plateforme IMDb.

Vous allez commencer par proposer une analyse complète de la base de données (Quels sont les acteurs les plus présents ? À quelle période ? La durée moyenne des films s'allonge ou se raccourcit avec les années ? Les acteurs de série sont-ils les mêmes qu'au cinéma ? Les acteurs ont en moyenne quel âge ? Quels sont les films les mieux notés ? Partagent-ils des caractéristiques communes ? etc...) Suite à une première analyse, vous pouvez décider de spécialiser votre cinéma, par exemple sur la « *période années 90* », ou alors sur « *les films d'action et d'aventure* », afin d'affiner votre exploration.

Après cette étape analytique, sur la fin du projet, vous utiliserez des algorithmes de machine learning pour recommander des films en fonction de films qui ont été appréciés par le spectateur.

Le client vous fournit également une base de données complémentaires venant de [TMDB](#), contenant des données sur les pays des boîtes de production, le budget, les recettes et également un chemin vers les posters des films. Il vous est demandé de

[récupérer les images](#) des films pour les afficher dans votre interface de recommandation.

**ATTENTION**, l'objectif n'est pas de diffuser dans le cinéma les films recommandés. L'objectif final est d'avoir une application avec d'une part des KPI et d'autre part le système de recommandation avec une zone de saisie de nom de film pour l'utilisateur. Cette application sera mise à disposition des clients du cinéma afin de leur proposer un service supplémentaire, en ligne, en plus du cinéma classique.

# Sources de données

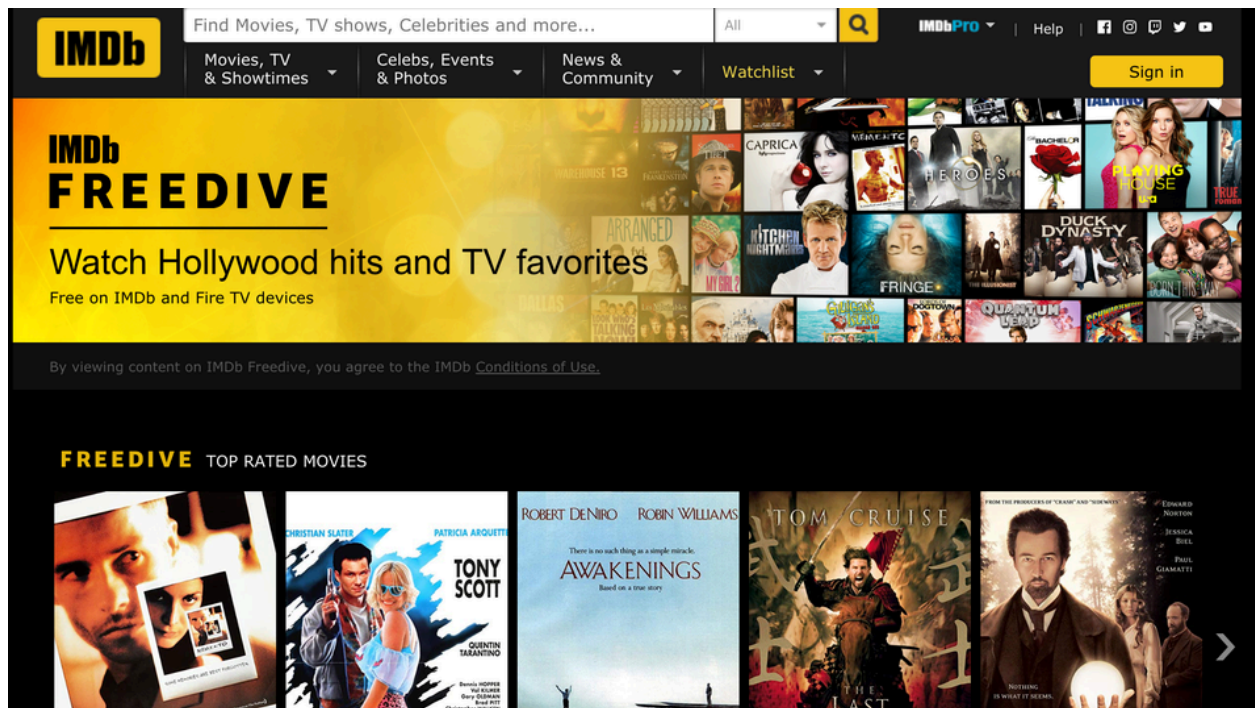
Les données sont disponibles sur le site IMDB. Les données sont réparties en plusieurs tables (les films, les acteurs, les réalisateurs, les notes, etc...)

- La documentation expliquant brièvement la définition de chaque colonne et de chaque table [est disponible ici](#).
- Les datasets [sont disponibles ici](#).
- Le dataset complémentaire est [disponible ici](#)

Remarques techniques :

- Vous pouvez télécharger les datasets en local, sur votre Drive ou bien sur un GitHub. Mais vous pouvez surtout ne pas les télécharger, et importer directement les datasets dans Pandas en mettant le lien du dataset.
- Les datasets sont très volumineux, il y a plus de 7M films et 10M acteurs référencés. Vous n'aurez sans doute pas besoin de la base complète. Une fois que vous aurez fait du nettoyage et des filtres sur ce que vous trouvez pertinent, pensez à exporter vos données "allégées". Ce sera plus rapide à réimporter.
- Pour rappel, Google Colab propose des serveurs "partagés". Les performances dépendent donc du nombre de personnes connectées en même temps. Parfois, vous ne pourrez donc pas charger tous ces volumineux datasets. N'hésitez pas à les traiter en local grâce à Anaconda / Jupyter.
- Les datasets IMDB sont au format TSV, pour "Tabulation Separated Values". C'est similaire au format CSV, mais séparé par des tabulations plutôt que des virgules. Vous pouvez utiliser la fonction suivante, qui indique que le séparateur est une tabulation :

```
pd.read_csv("dataset_link", sep = "\t", nrows=1000)
```



## Organisation et planning

Vous aurez besoin de faire des jointures (comme en SQL) entre les datasets, des graphiques en Python, des retraitements avec Pandas, du machine learning. Bien entendu, vous ne pourrez pas tout faire la première semaine, car vous apprendrez ces notions en parallèle du projet. Afin de vous donner de la visibilité, voici un planning indicatif, mais libre à vous de vous organiser :

- Semaine 1 et 2 : Appropriation et première exploration des données
  - Outils principaux : Pandas, Matplotlib, Seaborn
- Semaine 3 et 4 : Jointures, filtres, nettoyage, recherche de corrélation
  - Outils principaux : Pandas, Seaborn, Plotly
- Semaine 5 et 6 : Machine learning, recommandations
  - Outils principaux : scikit-learn
- Semaine 7 : Affinage, interface, présentation et Demo Day

## Rendu final

Le client aurait souhaité intégrer votre analyse et vos recommandations à son site pour pouvoir le tester, mais le timing est trop serré. Force de proposition, vous lui proposer de **le rendre testable au moyen d'un outil de votre choix.**

Le client a 2 besoins, qui peuvent être dans 2 outils séparés :

- Obtenir quelques statistiques sur les films (type, durée), acteurs (nombre de films, type de films) et d'autres. Vous le ferez notamment à l'aide de visualisations. Vous pouvez utiliser un outil de business intelligence, ou des graphiques via Python.
- Retourner une liste de films recommandés en fonction d'IDs ou de noms de films choisis par un utilisateur. Vous pouvez intégrer ces recommandations à un outil de dashboarding, ou bien faire ces recommandations directement depuis la ligne de commande ("input").

L'objectif n'est pas d'arriver à un travail parfait, mais que le système fonctionne et que vous arriviez à déceler les points à améliorer.

# Demo day

Vous expliquerez votre démarche et les difficultés rencontrés lors d'une **présentation orale de 10 à 15 minutes (à préciser avec le client)**, **et** ferez une **démonstration** devant des représentants des clients. Votre client vous donnera quelques noms de films pendant la présentation, vous effectuerez des recommandations de films basés sur ces films. Les autres spectateurs pourront aussi proposer un film.

## Missions

- Faire une présentation pour présenter votre travail, expliquer votre démarche, les outils utilisés, les difficultés rencontrées, et des pistes d'amélioration.
- Présenter les indicateurs statistiques et KPI pertinents sur les films.
- Créer un système de recommandation de film en utilisant des algorithmes de machine learning et faire une démonstration de ces recommandations sur des films proposés en séance par le client.

## Livrables attendus

- Un notebook contenant l'exploration et le nettoyage des données ainsi que les visualisations. Vous expliquerez vos choix de nettoyage et vos conclusions d'exploration dans un document de votre choix.
- Un dashboard présentant les KPI pertinents
- Un notebook pour l'étape Système de recommandation avec le code source avec vos commentaires

## Ressources

- <https://blogs.gartner.com/martin-kihn/how-to-build-a-recommender-system-in-python/>
- <https://linuxfr.org/news/predire-la-note-d-un-journal-sur-linuxfr-org>
- <https://www.mediego.com/fr/blog/netflix-success-story-basee-sur-algorithmes-de-recommandation/>
- <https://www.podcastscience.fm/dossiers/2012/04/25/les-algorithmes-de-recommandation/>



# Documentation

## IMDb Dataset Details

Each dataset is contained in a gzipped, tab-separated-values (TSV) formatted file in the UTF-8 character set. The first line in each file contains headers that describe what is in each column. A '\N' is used to denote that a particular field is missing or null for that title/name. The available datasets are as follows:

title.akas.tsv.gz - Contains the following information for titles:

- titleId (string) - a tconst, an alphanumeric unique identifier of the title
- ordering (integer) - a number to uniquely identify rows for a given titleId
- title (string) - the localized title
- region (string) - the region for this version of the title
- language (string) - the language of the title
- types (array) - Enumerated set of attributes for this alternative title. One or more of the following: "alternative", "dvd", "festival", "tv", "video", "working", "original", "imdbDisplay". New values may be added in the future without warning
- attributes (array) - Additional terms to describe this alternative title, not enumerated
- isOriginalTitle (boolean) - 0: not original title; 1: original title

title.basics.tsv.gz - Contains the following information for titles:

- tconst (string) - alphanumeric unique identifier of the title
- titleType (string) - the type/format of the title (e.g. movie, short, tvseries, tvepisode, video, etc)
- primaryTitle (string) - the more popular title / the title used by the filmmakers on promotional materials at the point of release
- originalTitle (string) - original title, in the original language
- isAdult (boolean) - 0: non-adult title; 1: adult title
- startYear (YYYY) - represents the release year of a title. In the case of TV Series, it is the series start year
- endYear (YYYY) - TV Series end year. '\N' for all other title types
- runtimeMinutes - primary runtime of the title, in minutes
- genres (string array) - includes up to three genres associated with the title

title.crew.tsv.gz - Contains the director and writer information for all the titles in IMDb.

Fields include:

- tconst (string) - alphanumeric unique identifier of the title
- directors (array of nconsts) - director(s) of the given title
- writers (array of nconsts) - writer(s) of the given title

title.episode.tsv.gz - Contains the tv episode information. Fields include:

- tconst (string) - alphanumeric identifier of episode
- parentTconst (string) - alphanumeric identifier of the parent TV Series
- seasonNumber (integer) - season number the episode belongs to
- episodeNumber (integer) - episode number of the tconst in the TV series

title.principals.tsv.gz - Contains the principal cast/crew for titles

- tconst (string) - alphanumeric unique identifier of the title
- ordering (integer) - a number to uniquely identify rows for a given titleId
- nconst (string) - alphanumeric unique identifier of the name/person
- category (string) - the category of job that person was in

- job (string) - the specific job title if applicable, else '\N'
- characters (string) - the name of the character played if applicable, else '\N'

title.ratings.tsv.gz – Contains the IMDb rating and votes information for titles

- tconst (string) - alphanumeric unique identifier of the title
- averageRating – weighted average of all the individual user ratings
- numVotes - number of votes the title has received

name.basics.tsv.gz – Contains the following information for names:

- nconst (string) - alphanumeric unique identifier of the name/person
- primaryName (string)– name by which the person is most often credited
- birthYear – in YYYY format
- deathYear – in YYYY format if applicable, else '\N'
- primaryProfession (array of strings)– the top-3 professions of the person
- knownForTitles (array of tconsts) – titles the person is known for