# Projet Movies9000 - ML

## 1. Introduction

Ce projet est basé sur un jeu de données de films de 9837 lignes (9836 échantillons + le titre des films), réparties entre 9 colonnes, et disponible en open-source sur Kaggle : https://www.kaggle.com/datasets/disham993/9000-movies-dataset/data

Il a été mis en ligne pour, je cite : «bâtir un système de recommandations de films en utilisant des modèles de NLP [traitement du langage naturel] et de Machine Learning».

Le Dataset contient les variables suivantes :

- **Release_Date** : date de sortie du film.
- **Title** : titre du film.
- **Overview** : résumé succinct de l'intrigue du film.
- **Popularity** : indicateur important calculé par les développeurs de TMDB, basé sur le nombre de vues par jour, de votes par jour, d'utilisateurs le marquant comme « favori » ou dans leur liste de suivi, la date de sortie, et d'autres critères.
- **Vote_Count** : nombre total de votes reçus de la part des spectateurs.
- **Vote_Average** : note moyenne sur 10, calculée à partir du nombre de votes et du nombre de spectateurs.
- **Original_Language** : langue originale du film ; les versions doublées ne sont pas prises en compte.
- **Genre** : catégories (genres) dans lesquelles le film peut être classé.
- **Poster_Url** : URL de l'affiche du film.

## 2. Exploration

Faisons un petit tour d'horizon des différentes variables présentes :

```
#   Column             Non-Null Count      Dtype
--- ------             --------------      -----
0 Release_Date         9837 non-null       object
1 Title                9828 non-null       object
2 Overview             9828 non-null       object
3 Popularity           9827 non-null       float64
4 Vote_Count           9827 non-null       object
5 Vote_Average         9827 non-null       object
6 Original_Language    9827 non-null       object
7 Genre                9826 non-null       object
8 Poster_Url           9826 non-null       object
```

On constate que nous n'avons pour l'heure que la colonne "Popularity" de type numérique.

Nous changeons donc le type de :

_"Release_Date" en DateTime,
_"Vote_Count" en entier,
_"Vote_Average" en flottant.

Ensuite, nous regarderons les valeurs manquantes, puis la significativité des variables ainsi que les statistiques descriptives des colonnes concernées (en fonction de leur type) afin de décider quoi faire desdites valeurs manquantes.

Comme dt.datetime n'est pas un type Numpy-compatible, nous traitons la colonne individuellement. Après avoir échoué à appliquer le formatage DateTime directement à la colonne des dates, nous utiliserons la librairie **re** avec l'expression régulière **r"^\d{4}-\d{2}-\d{2}$"** afin de ne conserver que les données s'assimilant à des dates (et donc, pouvant être transformées au format DateTime.

Nous changeons dans la foulée les types de *Vote_Count* en entier et de *Vote_Average* en flottant.

# 3. Nettoyage des données

Originellement, nous trouvons les données manquantes suivantes :

Release_Date 0
Title 9
Overview 9
Popularity 10
Vote_Count 10
Vote_Average 10
Original_Language 10
Genre 11
Poster_Url 11

Les données manquantes étant d'environ 11/9900 = 0,111… % (donc insignifiant), nous décidons de les supprimer.

De plus, nous avons vérifié qu'il n'y avait pas de doublons.

# 4. Données temporelles et ajustements

De base, nous n'avions qu'une seule variable temporelle : *Release_Date*.
Pour plus de lisibilité des données par le modèle, et pour observer d'éventuels effets de saisonnalité dans les parutions des films, on séparera par la suite, dans la colonne "Release_Date" :

- l'année (*Release_Year*),
- le mois (*Release_Month*),
- le jour du mois (*Release_Day*),
- et le jour de la semaine (*Release_Weekday*).

Nous décidons ensuite de binariser les colonnes *Genre* et *Original_Language*.

Enfin, comme ce projet sera plus orienté Machine Learning que NLP (ce qui n'exclut pas un projet complémentaire), nous décidons de supprimer les variables *Overview*, *Poster_Url* et *Title*.