PROJET NETFLOOX

Kaëlig B.
Laurence B.
Saeedullah R.Z.

Fev 2024

Trello

Espace de travail Trello
Gratuit

Tableaux
Membres
Paramètres d'espace de travail

Vues de l'espace de travail
Tableur
Calendrier

Vos tableaux
Buzzing Flash - Netfloox
Semaine 5 Fév
Semaine 12 Fev

Buzzing Flash - Netfloox | Visible par l'espace de travail | Tableau | Planyway | Power-ups | Automatisation | Filtres | Partager

Trello
- Mise en place du projet — 30 janv. — LB KB SK
- Créer les activités sous Trello. — 30 janv. — LB
- Planning du projet
- Connaissance des data — 1 — 5 févr. - 13 févr.

+ Ajouter une carte

Github
- Connecter les membres de l équipe sur github. — 6 févr. - 6 févr. — LB KB SK

+ Ajouter une carte

Table SQL
- Télécharger - Dézipper les fichiers — 6 févr. - 6 févr. — 1 — KB
- Partage des fichiers — 6 févr. - 6 févr. — 1 — LB KB SK
- Explorer les datas — Cette carte est un modèle. — 2/3
- Une base de données alimentées

+ Ajouter une carte

Interface web
- Streamlit ? — 1 — 16 févr. — LB KB SK
- Un modèle relationnel de données

+ Ajouter une carte

Un modèle relationnel de données
- Explorer les datas — 2/3
- Créer base de données
- https://developer.imdb.com/non-commercial-datasets/

+ Ajouter une carte

Python code
- Un algorithme de prédiction de la popularité
- Un algorithme de recommandatio
- Ouvrir sous python — 31 janv.

+ Ajouter une carte

Ordre du jour :

Introduction

Modèles

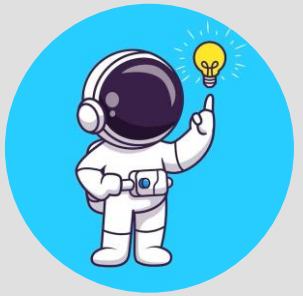
Résultats

Applications

Les besoins :

Vous passez plus de temps à choisir un film qu'à le regarder ?

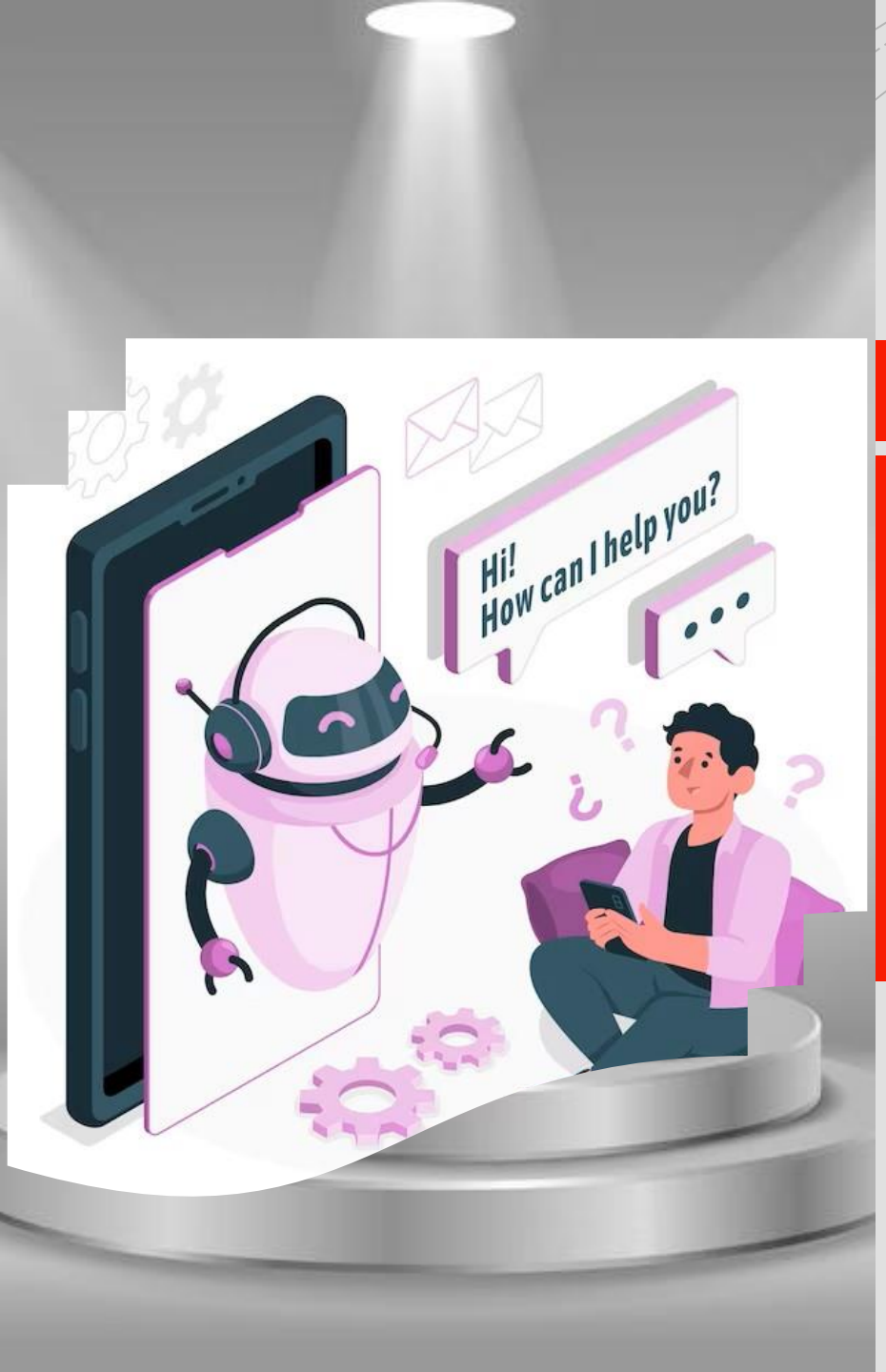
Les besoins :



Nous avons la solution !

Un service qui peut vous aider à faire le tri entre tous les contenus qui vous sont proposés et qui vous aidera à trouver les films et séries qui correspondent à vos critères.

« Popcorn Movies»

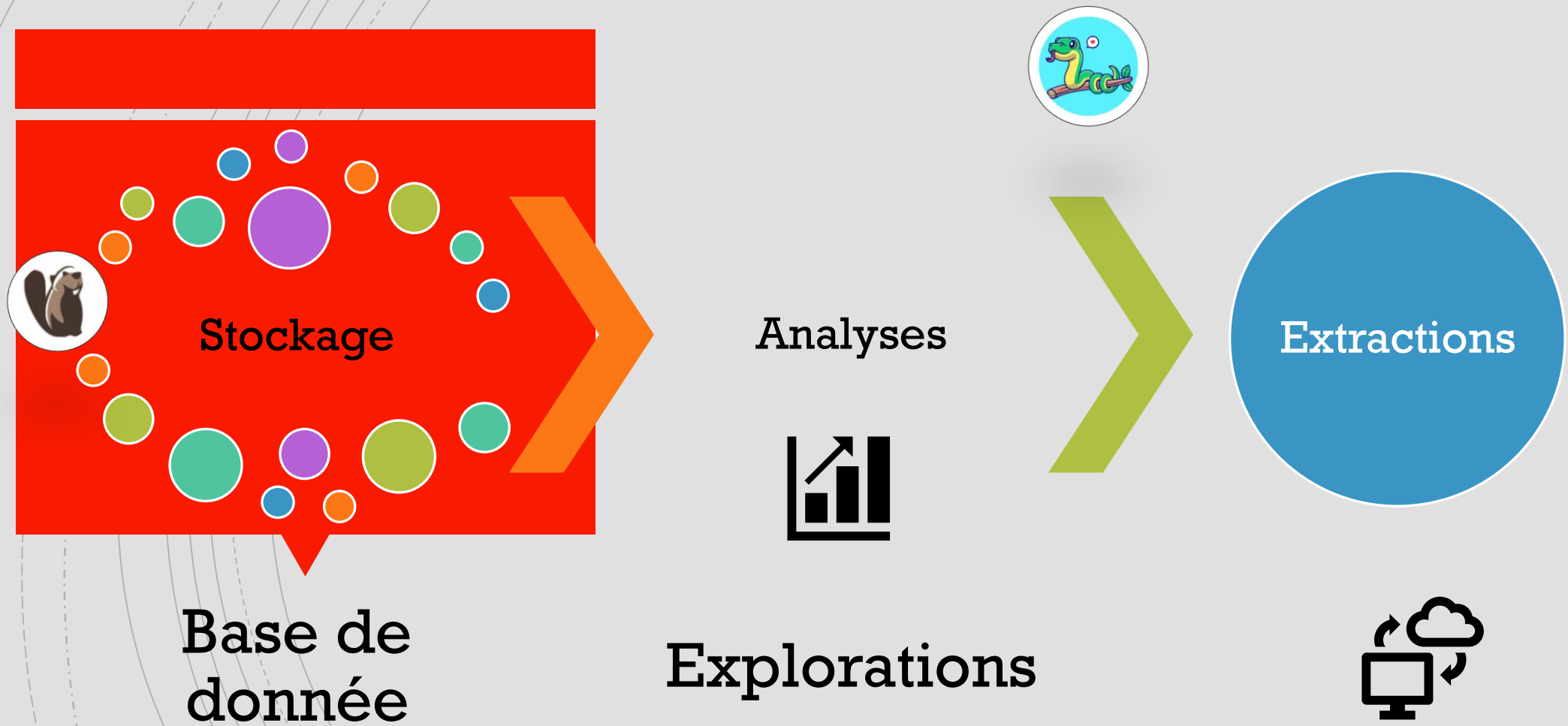


« Popcorn Movies»

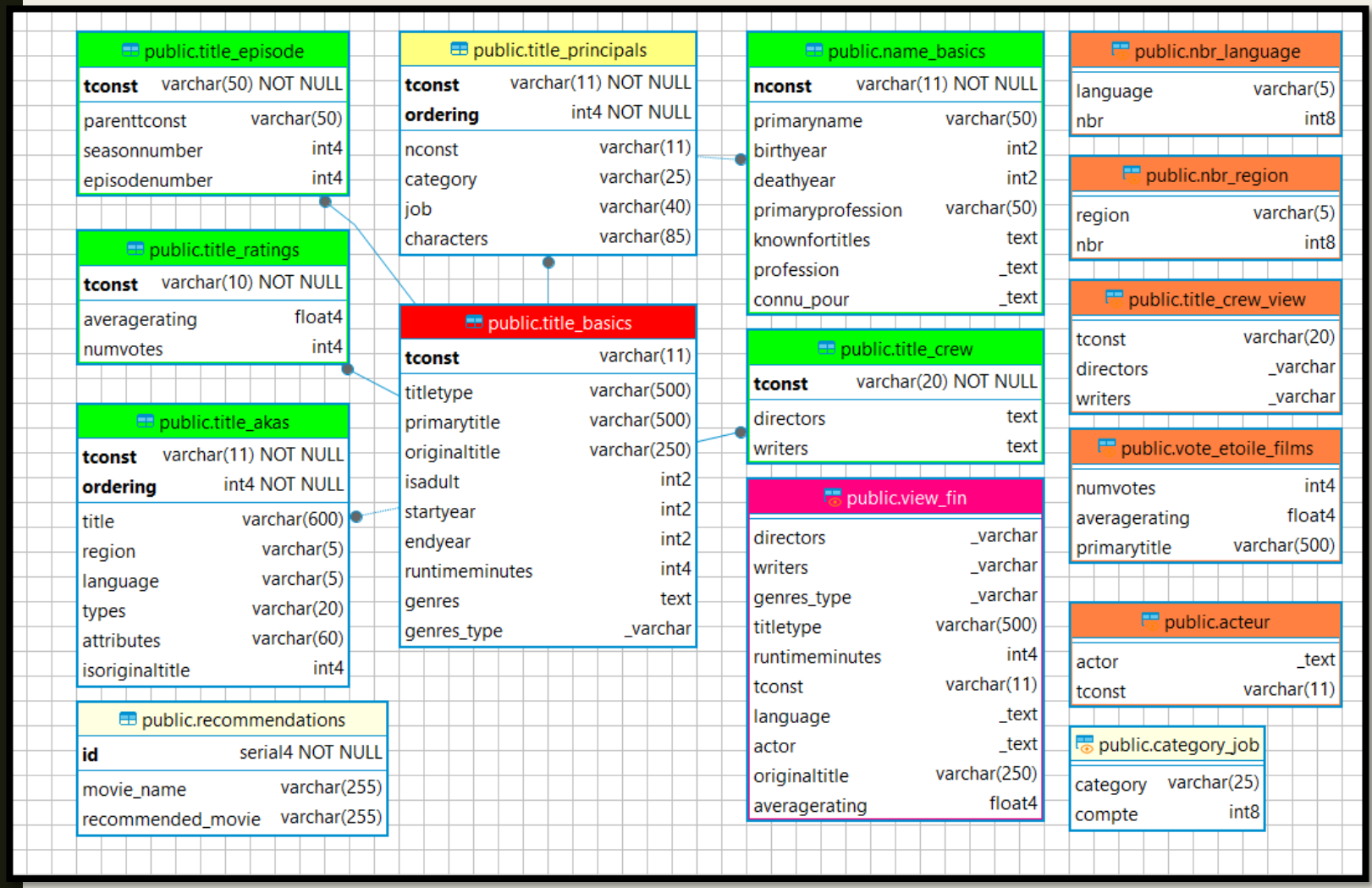
Il s’agit tout simplement d’un service de recommandation de films qui prend en compte vos préférences en matière de films.

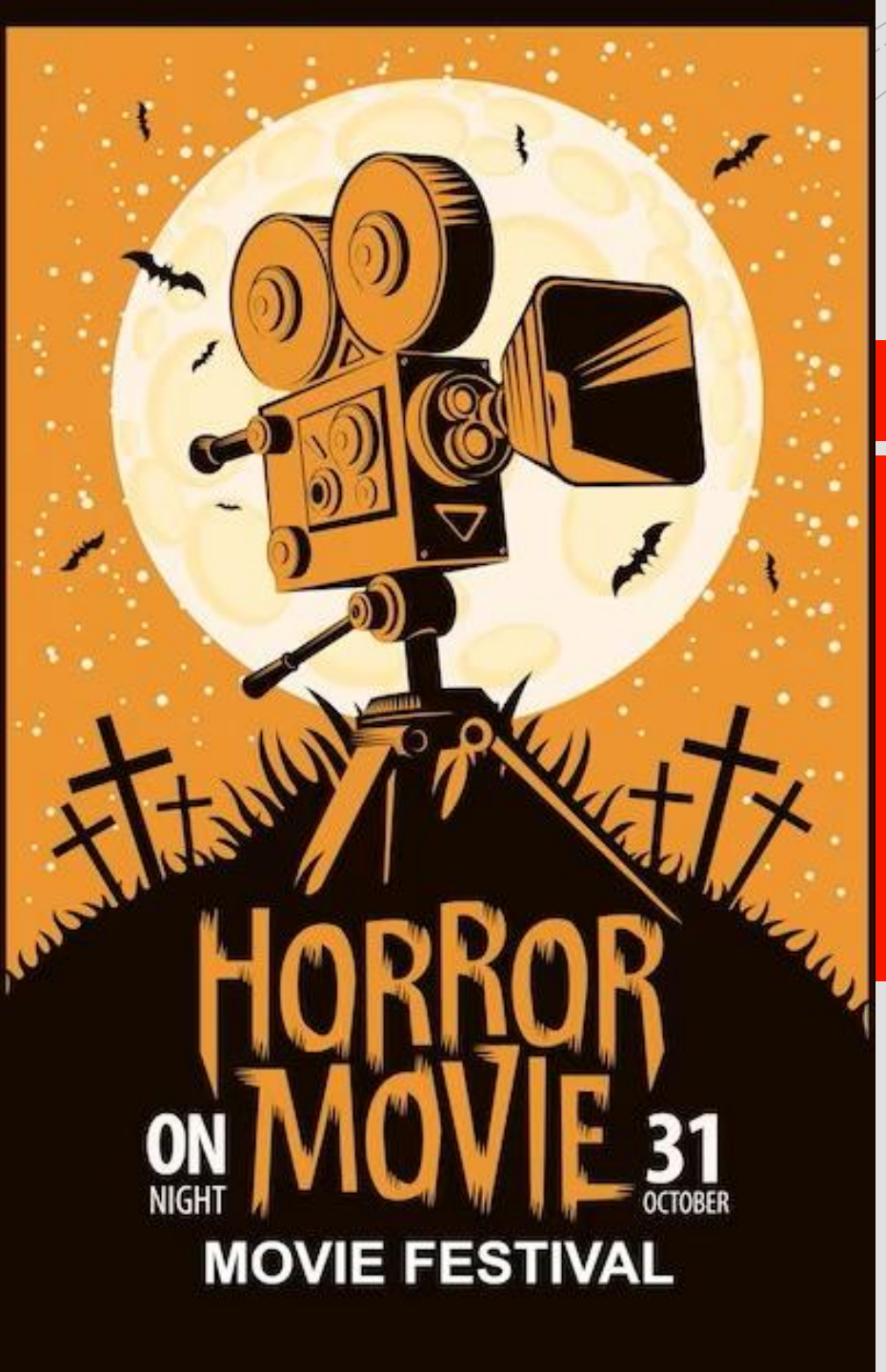
Recommandations personnalisées en fonction d’un film qui vous a plu.

Les données : 'Internet Movie Database (littéralement «Base de données cinématographiques d'Internet »).

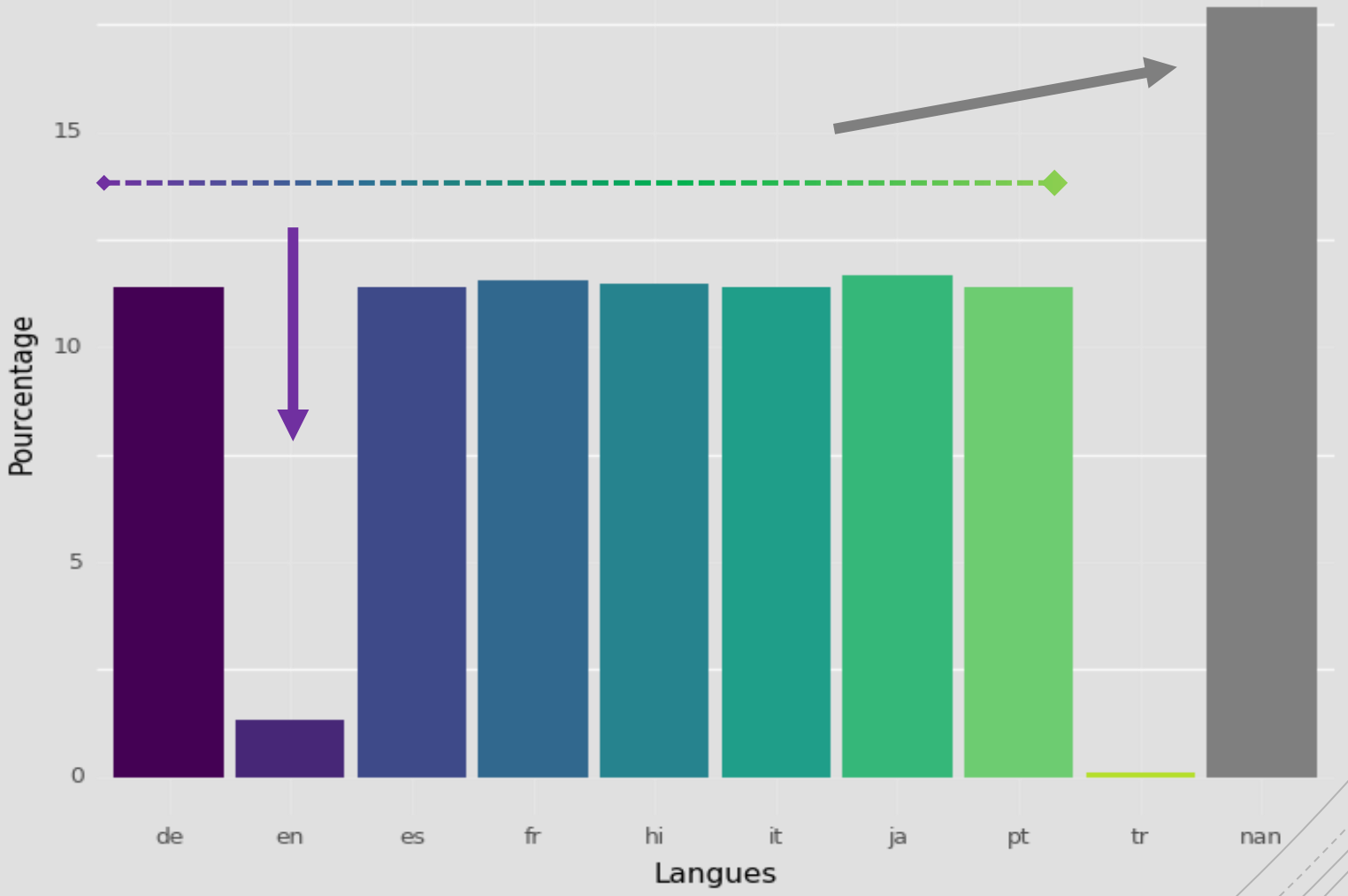


Base de données :



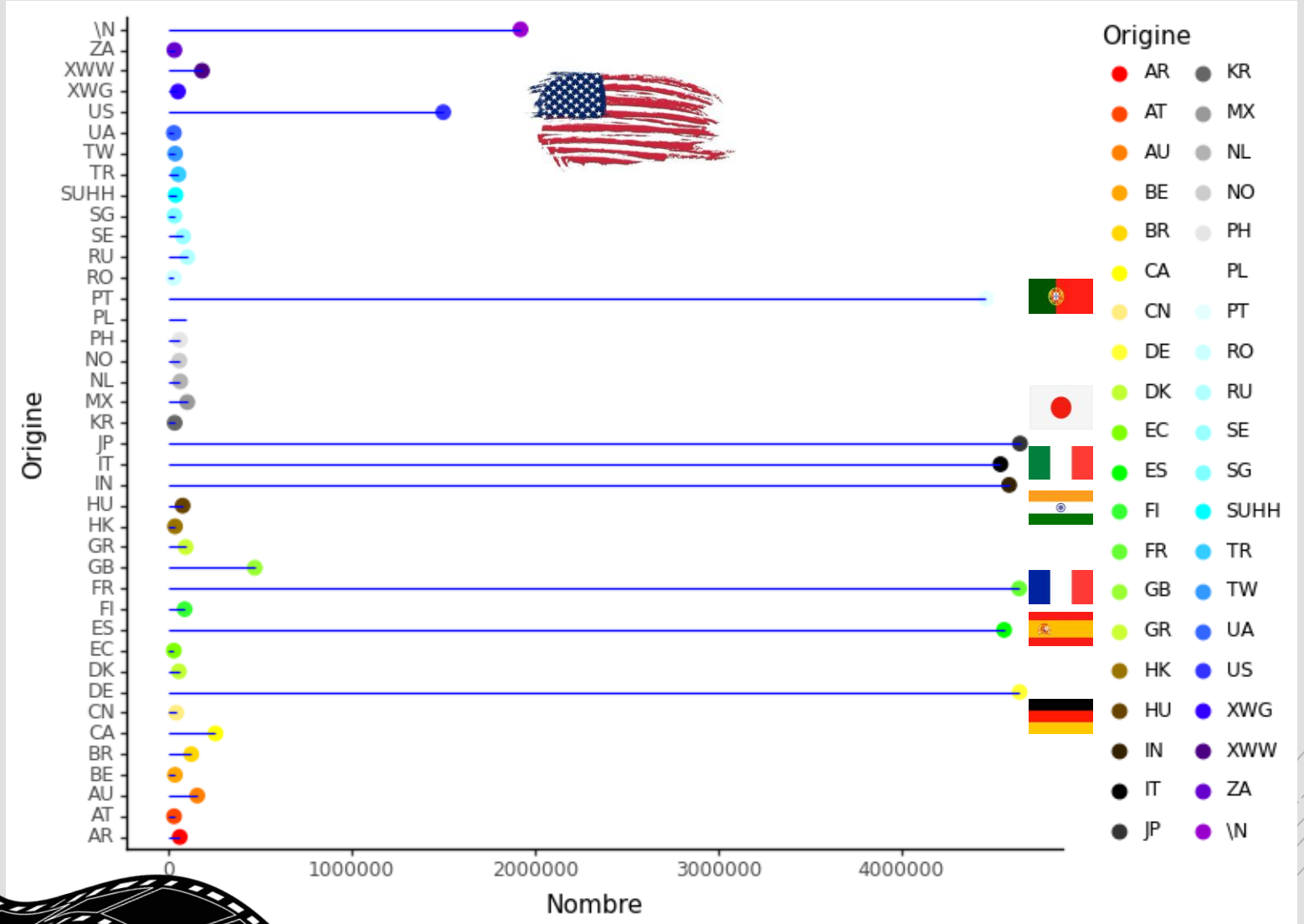


Top 10 des langues pour l'ensemble de la BD



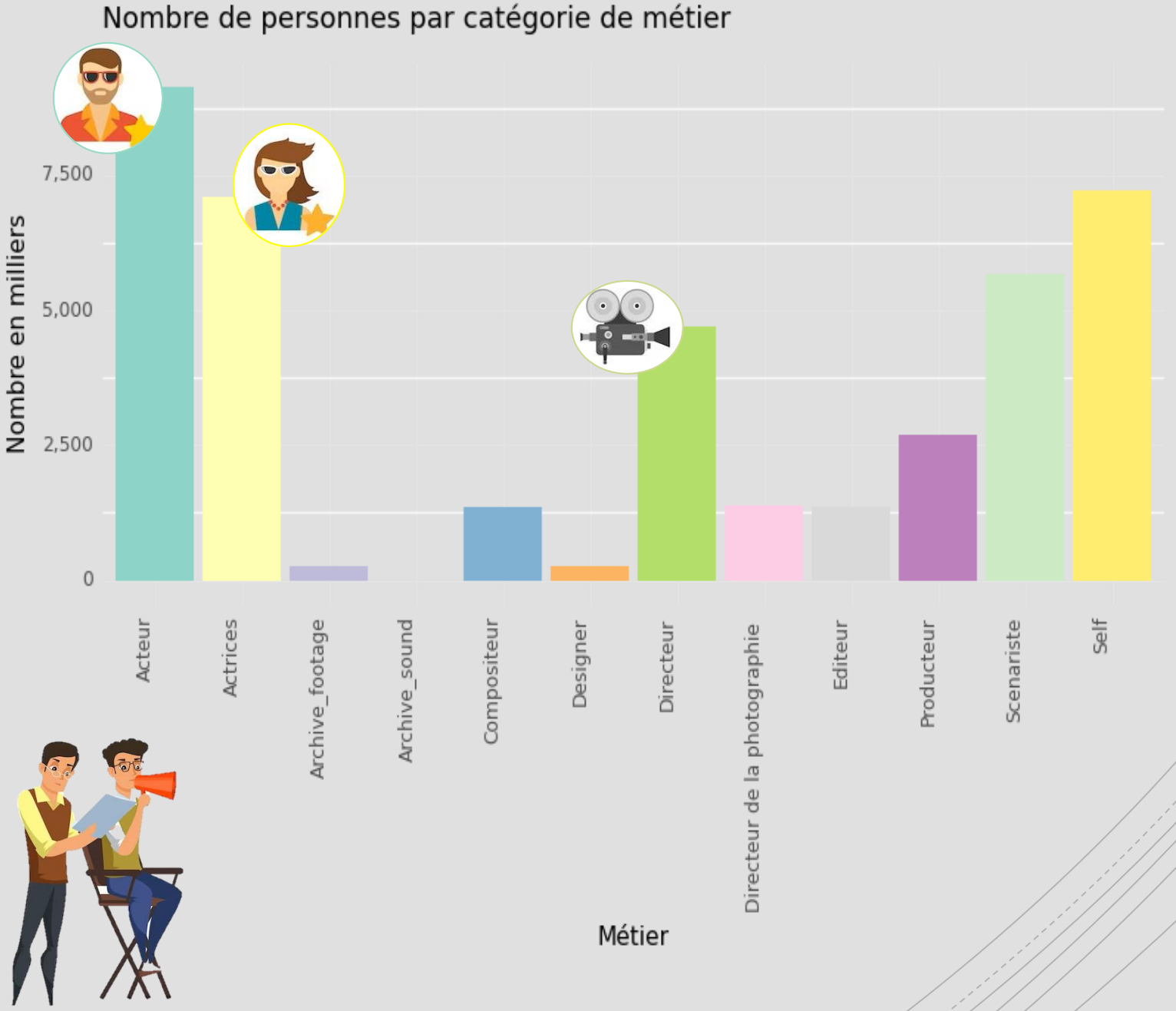
107 langues

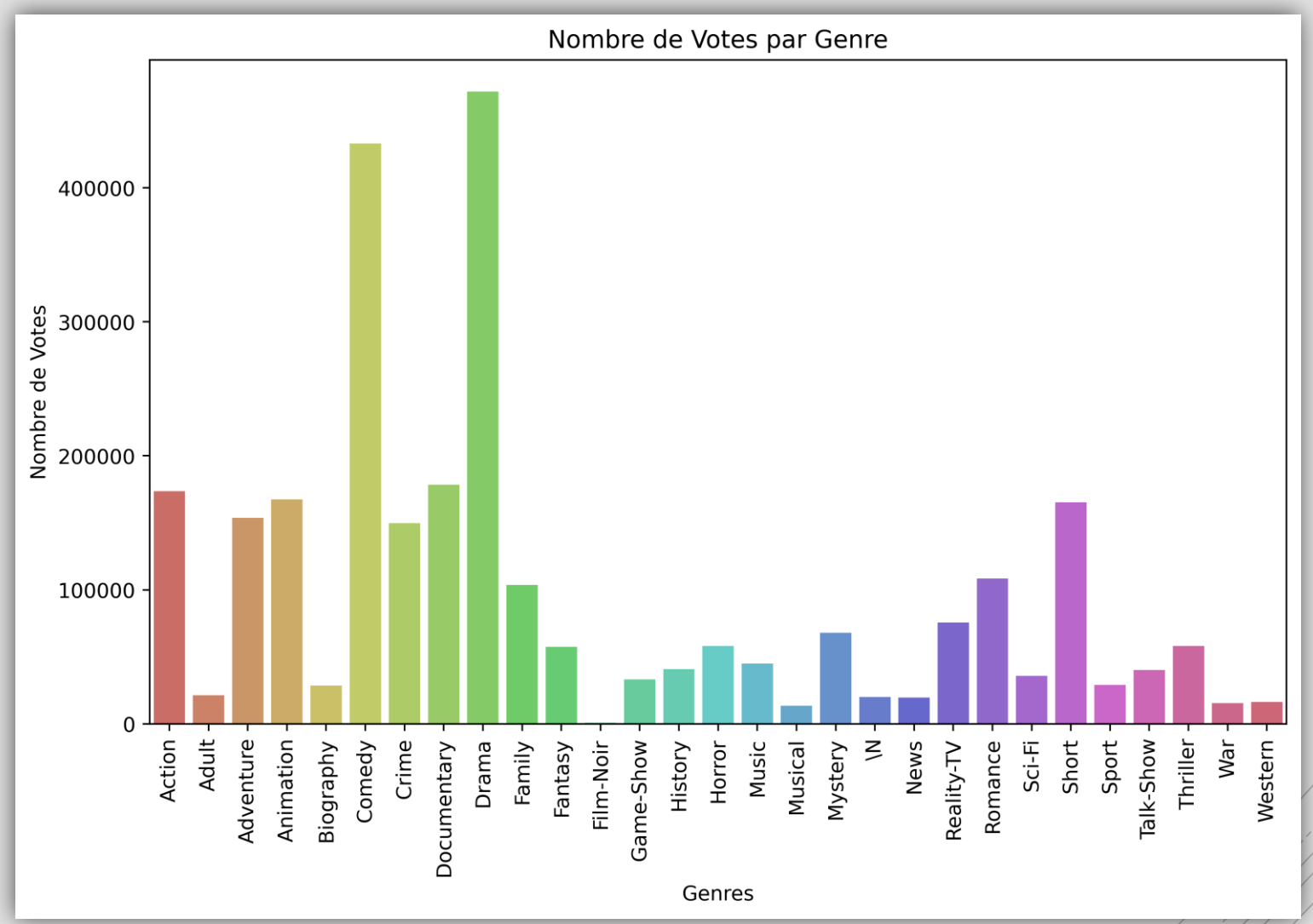
Top 40 de l'origine pour l'ensemble de la BD

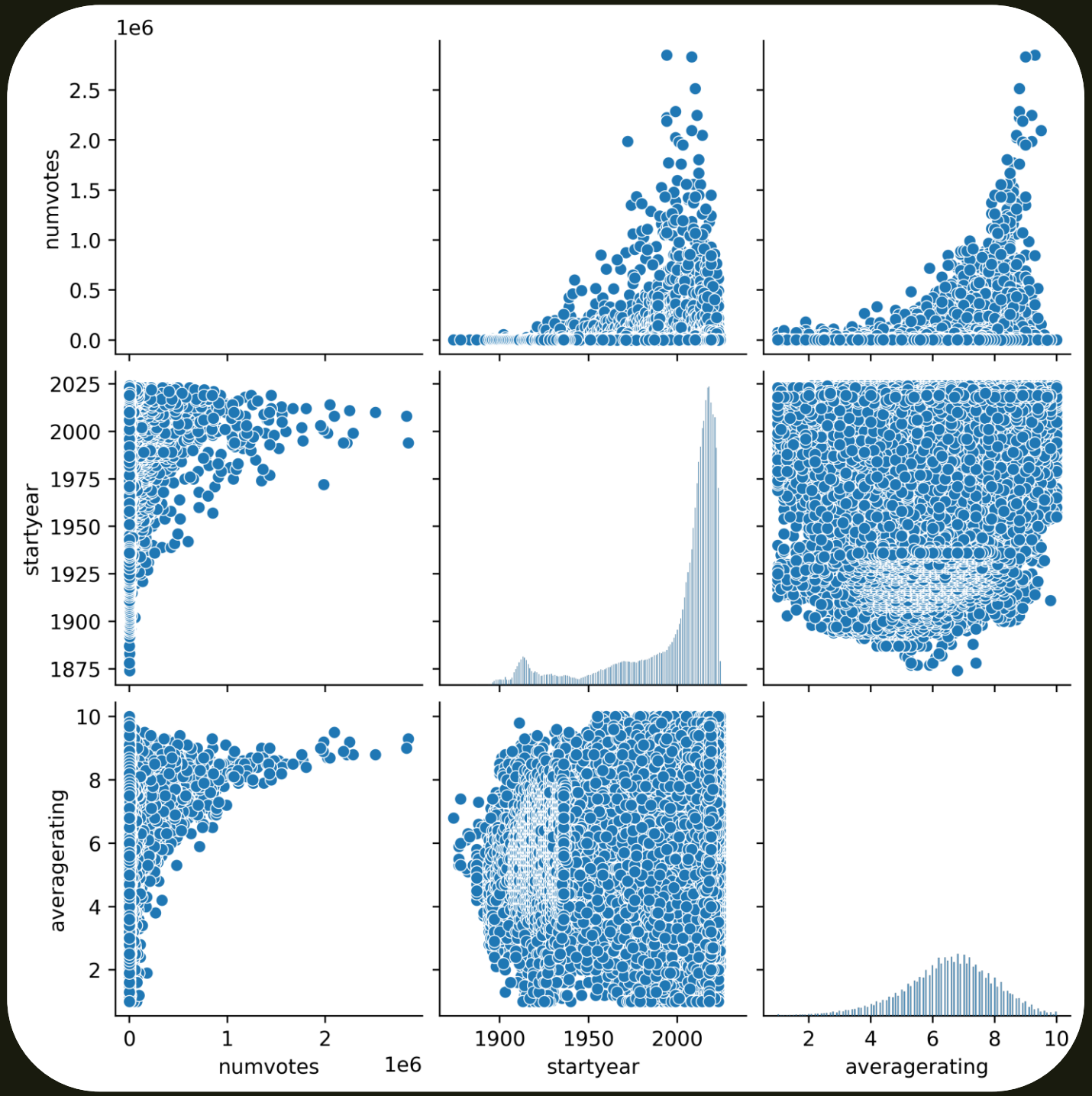


239 « régions »


POPC


Nombre de personnes par catégorie de métier
Nombre en milliers
7,500
5,000
2,500
0
Acteur
Actrices
Archive_footage
Archive_sound
Compositeur
Designer
Directeur
Directeur de la photographie
Editeur
Producteur
Scenariste
Self
Métier


Nombre de Votes par Genre
Nombre de Votes
0
100000
200000
300000
400000
Action
Adult
Adventure
Animation
Biography
Comedy
Crime
Documentary
Drama
Family
Fantasy
Film-Noir
Game-Show
History
Horror
Music
Musical
Mystery
\N
News
Reality-TV
Romance
Sci-Fi
Short
Sport
Talk-Show
Thriller
War
Western
Genres



Relation entre la notation, le nombre de vote et les années de production.

Modèle :

- BDD
- Extraction CSV
- Analyses
- Cosinus de similarité

Modèle :

Target

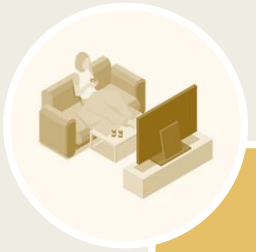
Original title

Features

- Directors
- Writers
- Genres
- Numvotes
- Actor
- StarYear
- Runtimesminutes

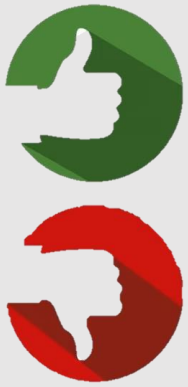
Recommandation

Recherche d’un film sur l’ensemble



Cosinus de similarity

Modèle de recommandation :



- Possible sur nos ordinateurs

- Re-calculer le cosinus à chaque film

- Pour éviter une matrice trop importante
- 5 films les + similaires

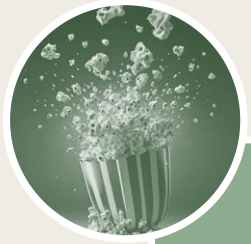
Modèle :

Target

Original title

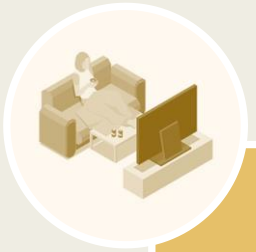
Features

- Directors
- Writers
- Genres
- Numvotes
- Actor
- StarYear
- Runtimesminutes



Prédiction

Analyses des points forts

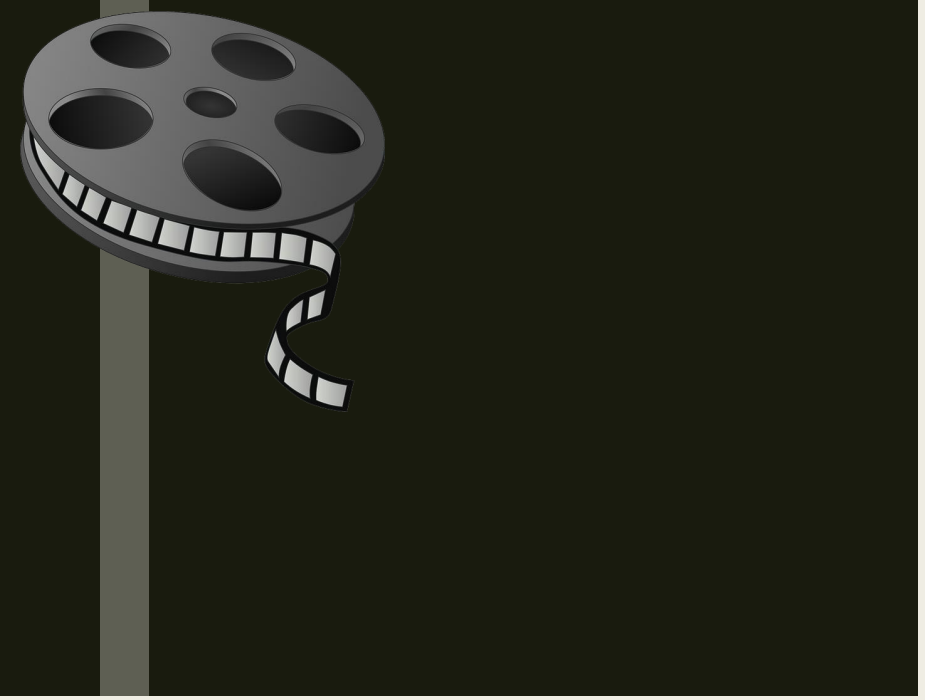


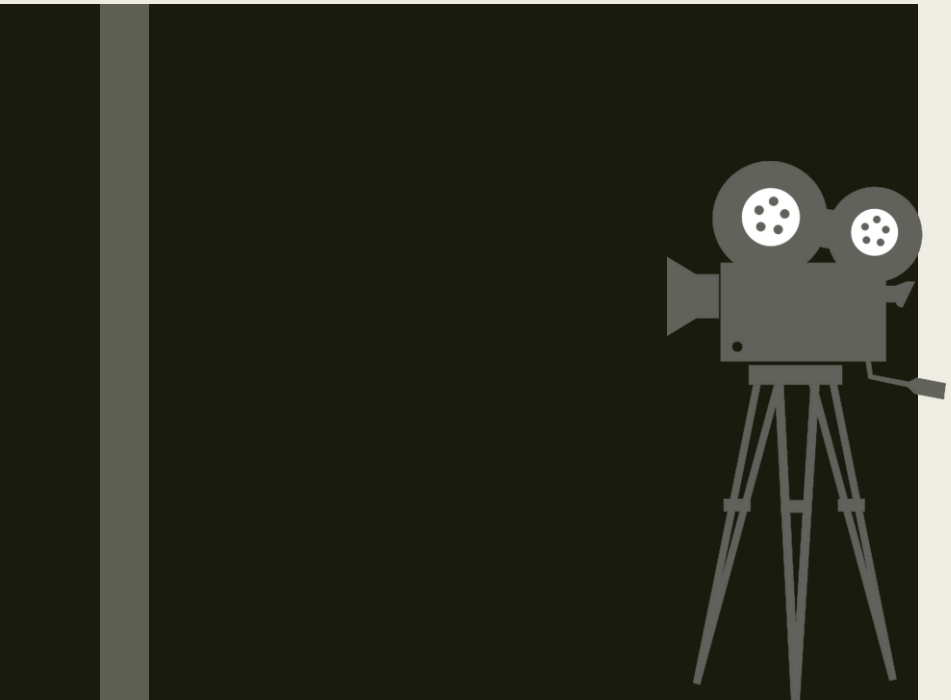
Random Forest

Pipeline

1 → 2 → 3

1. Séparation des features et de la target
2. Séparation des données quantitatives et qualitatives
3. Transformations

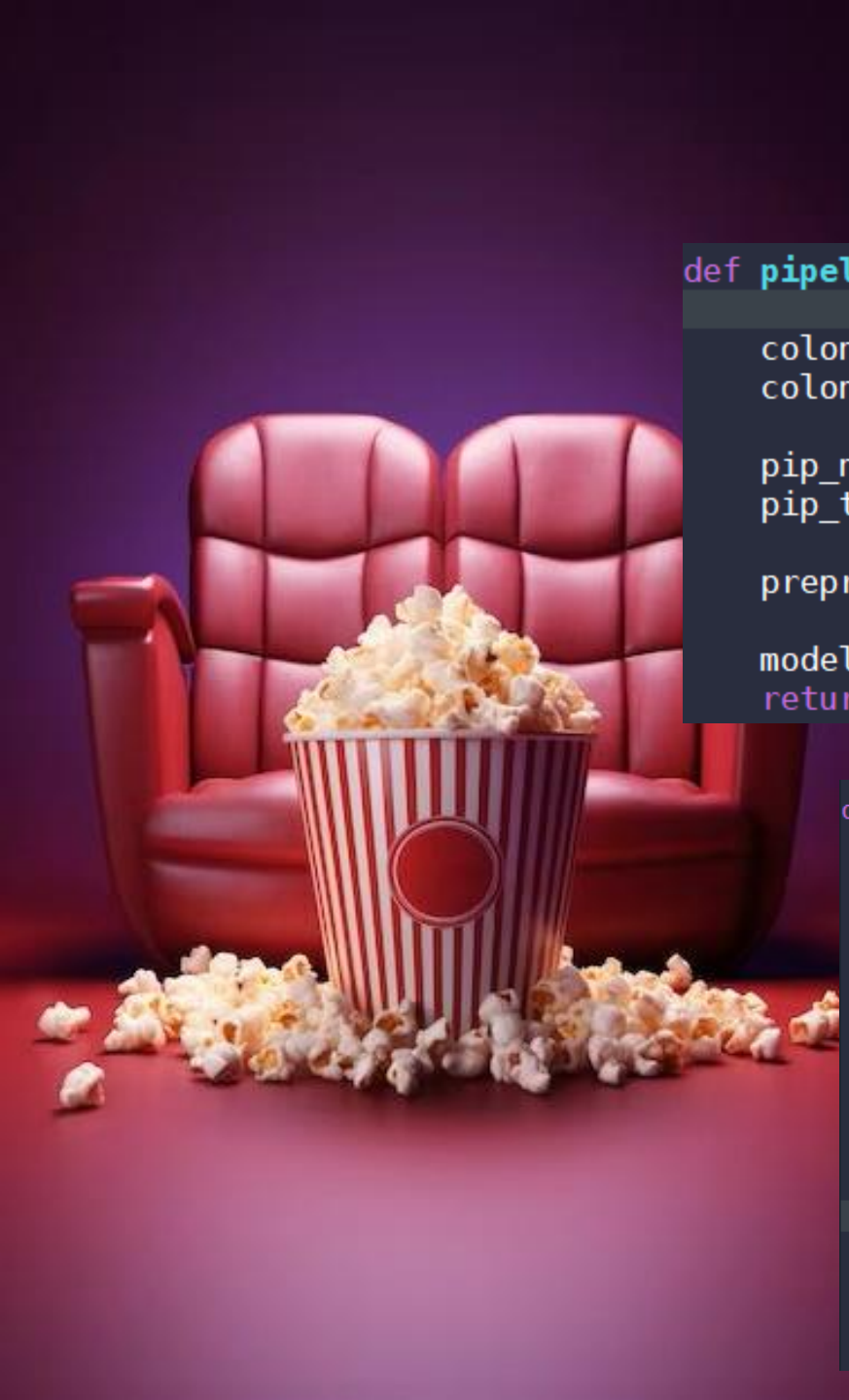




Comment entrainer le modèle

1 → 2 → 3 → 4

1	2	3	4
Pipeline	Modèle • Sélection	Amélioration • Grid Search	Sauvegarde



Modèle :

```python
def pipeline(model=DecisionTreeRegressor(), num=[SimpleImputer(),StandardScaler()],
             text=[OneHotEncoder(handle_unknown='ignore')]):
    colonne_numerique=['startyear','runtimeminutes','numvotes']
    colonne_nominal=[colonne for colonne in X.columns if colonne not in colonne_numerique]

    pip_num = Pipeline(steps=[(f"etape_num_{count}", i) for count, i in enumerate(num)])
    pip_text = Pipeline(steps=[(f"etape_text_{count}", i) for count, i in enumerate(text)])

    preprocess = ColumnTransformer(transformers=[('text', pip_text, colonne_nominal),
                                                 ('num', pip_num, colonne_numerique)])
    model = Pipeline(steps=[('preprocessor', preprocess), ('model', model)])
    return model
```

```python
def hGS(df,param,cv):
    y = df['averagerating']
    X = df.drop(['averagerating','titletype','features'],axis=1) #df['message']


    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)
    grid = HalvingGridSearchCV(estimator=pipeline(),param_grid=param,scoring='neg_mean_squared_error',
                               cv=cv,n_jobs=-1, verbose=1, error_score="raise")

    grid.fit(X_train, y_train)

    best_score = grid.best_score_
    best_params = grid.best_params_
    training_time = grid.cv_results_['mean_fit_time'].mean()
    return({'best_score': best_score,
            'best_params': best_params,
            'training_time': training_time,
            'fitted_model': grid.best_estimator_})
```

On regarde quoi ce soir ?

MERCI