DataFrame d'informations `df_info`

Introduction

Ce notebook a pour objectif de construire un DataFrame nommé `df_info`, qui regroupera toutes les informations relatives aux films (titre, genre, durée, année, etc.).

Ce DataFrame servira de base pour afficher les informations des films dans l'interface Streamlit.

Chargement des Datasets

In [1]:
```
import pandas as pd
```

Dataset `Machine Learning`

- Obtenir la liste des films contenus dans l'algorithme du machine learning

In [2]:
```
# Charger le fichier compressé
df_ml = pd.read_csv('../../machine learning/DF_ML.csv.gz', compression='gzip')

df_ml.head()
```

Out[2]:

	tconst	startYear	runtimeMinutes	Action	Adventure	Animation	Biography	Co
0	tt0000009	1894.0	45.0	False	False	False	False	
1	tt0000147	1897.0	100.0	False	False	False	False	
2	tt0000502	1905.0	100.0	False	False	False	False	
3	tt0000574	1906.0	70.0	True	True	False	True	
4	tt0000591	1907.0	90.0	False	False	False	False	

5 rows × 69 columns

In [3]:
```
# Garder uniquement les colonnes qui ne sont pas de type booléen
df_ml = df_ml.select_dtypes(exclude=['bool'])

# Vérification
df_ml.head()
```

Out[3]:

	tconst	startYear	runtimeMinutes	title_ratings_numVotes	title	tmdb_popu
0	tt0000009	1894.0	45.0	215.0	Miss Jerry	
1	tt0000147	1897.0	100.0	539.0	The Corbett-Fitzsimmons Fight	
2	tt0000502	1905.0	100.0	18.0	Bohemios	
3	tt0000574	1906.0	70.0	940.0	The Story of the Kelly Gang	
4	tt0000591	1907.0	90.0	28.0	L'enfant prodigue	

In [4]:
```
# Renommer les colonnes en français
df_ml = df_ml.rename(columns={
    'startYear': 'Année de Sortie',
    'runtimeMinutes': 'Durée (min)',
    'title_ratings_numVotes': 'Nombre Votes',
    'title': 'Titre',
    'tmdb_popularity': 'Popularité',
    'rating': 'Indice Bechdel',
    'notes': 'Note'
})

# Vérification des nouvelles colonnes
print(df_ml.columns)
```

```
Index(['tconst', 'Année de Sortie', 'Durée (min)', 'Nombre Votes', 'Titre',
       'Popularité', 'Indice Bechdel', 'nconst', 'Note'],
      dtype='object')
```

In [5]:
```
# Liste des colonnes dans l'ordre désiré
nouvel_ordre = [
                'tconst', 'Titre', 'Année de Sortie',
                'Durée (min)', 'Note', 'Nombre Votes',
                'Popularité', 'Indice Bechdel', 'nconst'
                ]

# Réorganisation du DataFrame
df_ml = df_ml[nouvel_ordre]

# Vérification
df_ml.head()
```

Out[5]:

	tconst	Titre	Année de Sortie	Durée (min)	Note	Nombre Votes	Popularité	Indice Bechdel	
0	tt0000009	Miss Jerry	1894.0	45.0	5.400000	215.0	0.000	0.0	nm0(
1	tt0000147	The Corbett-Fitzsimmons Fight	1897.0	100.0	5.211664	539.0	0.958	0.0	nm0
2	tt0000502	Bohemios	1905.0	100.0	4.400000	18.0	0.000	0.0	nm0(
3	tt0000574	The Story of the Kelly Gang	1906.0	70.0	5.981921	940.0	1.672	1.0	nm0
4	tt0000591	L'enfant prodigue	1907.0	90.0	5.700000	28.0	0.600	0.0	nm0

In [6]:
```
# Vérification des valeurs manquantes
df_ml.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 301086 entries, 0 to 301085
Data columns (total 9 columns):
 #   Column           Non-Null Count   Dtype  
---  ------           --------------   -----  
 0   tconst           301086 non-null  object 
 1   Titre            301086 non-null  object 
 2   Année de Sortie  301086 non-null  float64
 3   Durée (min)      301086 non-null  float64
 4   Note             301086 non-null  float64
 5   Nombre Votes     301086 non-null  float64
 6   Popularité       301086 non-null  float64
 7   Indice Bechdel   301086 non-null  float64
 8   nconst           269974 non-null  object 
dtypes: float64(6), object(3)
memory usage: 20.7+ MB
```

In [7]:
```
# Convertir les colonnes en entier
df_ml['Année de Sortie'] = df_ml['Année de Sortie'].fillna(0).astype(int)
df_ml['Durée (min)'] = df_ml['Durée (min)'].fillna(0).astype(int)
df_ml['Nombre Votes'] = df_ml['Nombre Votes'].fillna(0).astype(int)

df_ml.head()
```

Out[7]:

	tconst	Titre	Année de Sortie	Durée (min)	Note	Nombre Votes	Popularité	Indice Bechdel	
0	tt0000009	Miss Jerry	1894	45	5.400000	215	0.000	0.0	nm0(
1	tt0000147	The Corbett-Fitzsimmons Fight	1897	100	5.211664	539	0.958	0.0	nm0
2	tt0000502	Bohemios	1905	100	4.400000	18	0.000	0.0	nm0(
3	tt0000574	The Story of the Kelly Gang	1906	70	5.981921	940	1.672	1.0	nm0
4	tt0000591	L'enfant prodigue	1907	90	5.700000	28	0.600	0.0	nm0

Dataset `Title.Basics`

- Récupérer les informations nécessaires pour `df_info`

In [8]:
```
title_basics = pd.read_csv('../../gitignore/title_basics_traite.csv')

title_basics.head()
```

Out[8]:

	tconst	titleType	startYear	runtimeMinutes	genres	decade	A
0	tt0000009	movie	1894	45	Romance	1890	
1	tt0000147	movie	1897	100	Documentary,News,Sport	1890	
2	tt0000502	movie	1905	100	\N	1900	
3	tt0000574	movie	1906	70	Action,Adventure,Biography	1900	
4	tt0000591	movie	1907	90	Drama	1900	

5 rows × 34 columns

In [9]:
```
# Garder uniquement les colonnes qui ne sont pas de type booléen
title_basics = title_basics.select_dtypes(exclude=['bool'])

# Vérification
title_basics.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 688341 entries, 0 to 688340
Data columns (total 6 columns):
 #   Column          Non-Null Count   Dtype
---  ------          --------------   -----
 0   tconst          688341 non-null  object
 1   titleType       688341 non-null  object
 2   startYear       688341 non-null  int64
 3   runtimeMinutes  688341 non-null  int64
 4   genres          688341 non-null  object
 5   decade          688341 non-null  int64
dtypes: int64(3), object(3)
memory usage: 31.5+ MB
```

In [10]:
```
# Vérification des valeurs manquantes
print(title_basics.isnull().sum())
```

```
tconst            0
titleType         0
startYear         0
runtimeMinutes    0
genres            0
decade            0
dtype: int64
```

In [11]:
```
# Comptage des occurrences de '\N' dans chaque colonne
for col in title_basics.columns:
    count_null_values = title_basics[col].eq(r'\N').sum()
    if count_null_values > 0:
        print(f"Colonne '{col}' contient {count_null_values} occurrences de '\\N
```

```
Colonne 'genres' contient 75116 occurrences de '\N'.
```

In [12]:
```
# Remplacement des occurrences de '\N' par NaN
title_basics = title_basics.replace(r'\N', 'Non renseigné')
```

Dataset `TmdB.Full`

- Obtenir les informations `Affiches` , `Url` , `Synopsis`

In [ ]:
```
tmdb = pd.read_csv('../../gitignore/tmdb_full.csv')

tmdb.head()
```

In [14]:
```
# Garder uniquement les colonnes qui ne sont pas de type booléen
tmdb = tmdb.select_dtypes(exclude=['bool'])

# Vérification
tmdb.head()
```

Out[14]:

	backdrop_path	budget	genres	homepage	id	imdb_i
0	/dvQj1GBZAZirz1skEEZyWH2ZqQP.jpg	0	['Comedy']	NaN	3924	tt002992
1	NaN	0	['Adventure']	NaN	6124	tt001143
2	/uJlc4aNPF3Y8yAqahJTKBwgwPVW.jpg	0	['Drama', 'Romance']	NaN	8773	tt005574
3	/hQ4pYsIbP22TMXOUdSfC2mjWrO0.jpg	0	['Drama', 'Comedy', 'Crime']	NaN	2	tt009467
4	/l94l89eMmFKh7na2a1u5q67VgNx.jpg	0	['Drama', 'Comedy', 'Romance']	NaN	3	tt009214

5 rows × 23 columns

In [15]:
```
tmdb.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 309572 entries, 0 to 309571
Data columns (total 23 columns):
 #   Column                        Non-Null Count   Dtype
---  ------                        --------------   -----
 0   backdrop_path                 151760 non-null  object
 1   budget                        309572 non-null  int64
 2   genres                        309572 non-null  object
 3   homepage                      44262 non-null   object
 4   id                            309572 non-null  int64
 5   imdb_id                       309572 non-null  object
 6   original_language             309572 non-null  object
 7   original_title                309572 non-null  object
 8   overview                      282512 non-null  object
 9   popularity                    309572 non-null  float64
 10  poster_path                   264159 non-null  object
 11  production_countries          309572 non-null  object
 12  release_date                  301339 non-null  object
 13  revenue                       309572 non-null  int64
 14  runtime                       309572 non-null  int64
 15  spoken_languages              309572 non-null  object
 16  status                        309572 non-null  object
 17  tagline                       74573 non-null   object
 18  title                         309572 non-null  object
 19  vote_average                  309572 non-null  float64
 20  vote_count                    309572 non-null  int64
 21  production_companies_name     309572 non-null  object
 22  production_companies_country  164438 non-null  object
dtypes: float64(2), int64(5), object(16)
memory usage: 54.3+ MB
```

In [16]:

```python
# Mettre les colonnes en français
colonnes_en_francais = {
    'backdrop_path': 'Chemin Affiche Pub',
    'budget': 'Budget',
    'genres': 'Genres',
    'homepage': 'Url Site',
    'id': 'Identifiant',
    'imdb_id': 'tconst',
    'original_language': 'Langue Originale',
    'original_title': 'Titre Original',
    'overview': 'Synopsis',
    'popularity': 'Popularité',
    'poster_path': 'Chemin Affiche',
    'production_countries': 'Pays Production',
    'release_date': 'Date Sortie',
    'revenue': 'Revenus',
    'runtime': 'Durée',
    'spoken_languages': 'Langues Parlées',
    'status': 'Statut',
    'tagline': 'Slogan',
    'title': 'Titre',
    'vote_average': 'Note',
    'vote_count': 'Nombre Votes',
    'production_companies_name': 'Maison de Production',
    'production_companies_country': 'Pays Maison Production'
}

# Renommer les colonnes
tmdb = tmdb.rename(columns=colonnes_en_francais)
```

```
# Vérification des nouvelles colonnes
print(tmdb.columns)
```

```
Index(['Chemin Affiche Pub', 'Budget', 'Genres', 'Url Site', 'Identifiant',
       'tconst', 'Langue Originale', 'Titre Original', 'Synopsis',
       'Popularité', 'Chemin Affiche', 'Pays Production', 'Date Sortie',
       'Revenus', 'Durée', 'Langues Parlées', 'Statut', 'Slogan', 'Titre',
       'Note', 'Nombre Votes', 'Maison de Production',
       'Pays Maison Production'],
      dtype='object')
```

In [17]:
```
# Export pour analyse KPI
tmdb.to_csv('../../gitignore/KPI_tmdb_export.csv', index=False, encoding='utf-8'
```

In [18]:
```
# Supprimer plusieurs colonnes
colonnes_a_supprimer = ['Budget', 'Genres', 'Identifiant', 'Popularité', 'Date S

tmdb = tmdb.drop(columns=colonnes_a_supprimer)

# Vérification des colonnes restantes
print(tmdb.columns)
```

```
Index(['Chemin Affiche Pub', 'Url Site', 'tconst', 'Langue Originale',
       'Titre Original', 'Synopsis', 'Chemin Affiche', 'Pays Production',
       'Langues Parlées', 'Statut', 'Slogan', 'Maison de Production'],
      dtype='object')
```

DataSet `name.basic_info_acteurs`

In [19]:
```
info_acteurs = pd.read_csv('../../gitignore/info_casting_acteurs.tsv', sep= '\t'

info_acteurs.head()
```

Out[19]:

	tconst	nconst	catégorie	rôle
0	tt0000001	nm1588970	self	soi-même
1	tt0000005	nm0443482	actor	Blacksmith
2	tt0000005	nm0653042	actor	Assistant
3	tt0000007	nm0179163	actor	NaN
4	tt0000007	nm0183947	actor	NaN

Merge des tables

- Lier les tables entre elles pour obtenir un seul `df_info`

Merge 1 : `df_ml` et `title_basics`

In [20]:
```
df_merge_1 = pd.merge(df_ml, title_basics, on='tconst', how='left')

df_merge_1.head()
```

Out[20]:

	tconst	Titre	Année de Sortie	Durée (min)	Note	Nombre Votes	Popularité	Indice Bechdel	
0	tt0000009	Miss Jerry	1894	45	5.400000	215	0.000	0.0	nm0(
1	tt0000147	The Corbett-Fitzsimmons Fight	1897	100	5.211664	539	0.958	0.0	nm0
2	tt0000502	Bohemios	1905	100	4.400000	18	0.000	0.0	nm0(
3	tt0000574	The Story of the Kelly Gang	1906	70	5.981921	940	1.672	1.0	nm0
4	tt0000591	L'enfant prodigue	1907	90	5.700000	28	0.600	0.0	nm0

In [21]:
```
# Supprimer plusieurs colonnes
colonnes_a_supprimer = ['startYear', 'runtimeMinutes']

df_merge_1 = df_merge_1.drop(columns=colonnes_a_supprimer)

# Vérification des colonnes restantes
print(df_merge_1.columns)
```

```
Index(['tconst', 'Titre', 'Année de Sortie', 'Durée (min)', 'Note',
       'Nombre Votes', 'Popularité', 'Indice Bechdel', 'nconst', 'titleType',
       'genres', 'decade'],
      dtype='object')
```

In [22]:
```
# Liste des colonnes dans l'ordre désiré
nouvel_ordre = [
                'tconst', 'Titre', 'genres', 'Année de Sortie',
                'Durée (min)', 'Note', 'Nombre Votes',
                'Popularité', 'decade', 'Indice Bechdel',
                'nconst', 'titleType',
                ]

# Réorganisation du DataFrame
df_merge_1 = df_merge_1[nouvel_ordre]

# Vérification
df_merge_1.head()
```

Out[22]:

	tconst	Titre	genres	Année de Sortie	Durée (min)	Note	Nombre Votes
0	tt0000009	Miss Jerry	Romance	1894	45	5.400000	215
1	tt0000147	The Corbett-Fitzsimmons Fight	Documentary,News,Sport	1897	100	5.211664	539
2	tt0000502	Bohemios	Non renseigné	1905	100	4.400000	18
3	tt0000574	The Story of the Kelly Gang	Action,Adventure,Biography	1906	70	5.981921	940
4	tt0000591	L'enfant prodigue	Drama	1907	90	5.700000	28

In [23]:
```
# Renommer les colonnes en français
df_merge_1 = df_merge_1.rename(columns={'decade': 'Décennie'})

# Vérification des nouvelles colonnes
print(df_merge_1.columns)
```

```
Index(['tconst', 'Titre', 'genres', 'Année de Sortie', 'Durée (min)', 'Note',
       'Nombre Votes', 'Popularité', 'Décennie', 'Indice Bechdel', 'nconst',
       'titleType'],
      dtype='object')
```

In [24]:
```
df_merge_1 = df_merge_1.drop(columns = 'titleType')
```

In [25]:
```
df_merge_1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 301086 entries, 0 to 301085
Data columns (total 11 columns):
 #   Column           Non-Null Count   Dtype  
---  ------           --------------   -----  
 0   tconst           301086 non-null  object 
 1   Titre            301086 non-null  object 
 2   genres           301086 non-null  object 
 3   Année de Sortie  301086 non-null  int32  
 4   Durée (min)      301086 non-null  int32  
 5   Note             301086 non-null  float64
 6   Nombre Votes     301086 non-null  int32  
 7   Popularité       301086 non-null  float64
 8   Décennie         301086 non-null  int64  
 9   Indice Bechdel   301086 non-null  float64
 10  nconst           269974 non-null  object 
dtypes: float64(3), int32(3), int64(1), object(4)
memory usage: 21.8+ MB
```

Merge 2 : df_merge_1 + tmdb

In [26]:
```
df_merge_2 = pd.merge(df_merge_1, tmdb, on='tconst', how='left')

df_merge_2.head()
```

Out[26]:

	tconst	Titre	genres	Année de Sortie	Durée (min)	Note	Nombre Votes
0	tt0000009	Miss Jerry	Romance	1894	45	5.400000	215
1	tt0000147	The Corbett-Fitzsimmons Fight	Documentary,News,Sport	1897	100	5.211664	539
2	tt0000502	Bohemios	Non renseigné	1905	100	4.400000	18
3	tt0000574	The Story of the Kelly Gang	Action,Adventure,Biography	1906	70	5.981921	940
4	tt0000591	L'enfant prodigue	Drama	1907	90	5.700000	28

5 rows × 22 columns

In [27]:
```
df_merge_2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 301086 entries, 0 to 301085
Data columns (total 22 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   tconst                301086 non-null  object
 1   Titre                 301086 non-null  object
 2   genres                301086 non-null  object
 3   Année de Sortie       301086 non-null  int32
 4   Durée (min)           301086 non-null  int32
 5   Note                  301086 non-null  float64
 6   Nombre Votes          301086 non-null  int32
 7   Popularité            301086 non-null  float64
 8   Décennie              301086 non-null  int64
 9   Indice Bechdel        301086 non-null  float64
 10  nconst                269974 non-null  object
 11  Chemin Affiche Pub    115058 non-null  object
 12  Url Site              29217 non-null   object
 13  Langue Originale      197925 non-null  object
 14  Titre Original        197925 non-null  object
 15  Synopsis              184658 non-null  object
 16  Chemin Affiche        180296 non-null  object
 17  Pays Production       197925 non-null  object
 18  Langues Parlées       197925 non-null  object
 19  Statut                197925 non-null  object
 20  Slogan                59740 non-null   object
 21  Maison de Production  197925 non-null  object
dtypes: float64(3), int32(3), int64(1), object(15)
memory usage: 47.1+ MB
```

In [28]:

```python
# Liste de l'ordre des colonnes souhaité
nouvel_ordre_colonnes = [
    'tconst', 'Titre', 'Titre Original', 'genres',
    'Année de Sortie', 'Durée (min)', 'Note',
    'Nombre Votes', 'Popularité', 'Décennie',
    'Indice Bechdel', 'Maison de Production',
    'Pays Production', 'nconst', 'Slogan',
    'Synopsis', 'Chemin Affiche', 'Chemin Affiche Pub',
    'Url Site', 'Langue Originale', 'Langues Parlées',
    'Statut',
]

# Réorganiser les colonnes dans l'ordre spécifié
df_merge_2 = df_merge_2[nouvel_ordre_colonnes]

# Vérification du résultat
df_merge_2.head()
```

Out[28]:

	tconst	Titre	Titre Original	genres	Année de Sortie	Durée (min)	Nc
0	tt0000009	Miss Jerry	NaN	Romance	1894	45	5.4000
1	tt0000147	The Corbett-Fitzsimmons Fight	The Corbett-Fitzsimmons Fight	Documentary,News,Sport	1897	100	5.2116
2	tt0000502	Bohemios	NaN	Non renseigné	1905	100	4.4000
3	tt0000574	The Story of the Kelly Gang	The Story of the Kelly Gang	Action,Adventure,Biography	1906	70	5.9819
4	tt0000591	L'enfant prodigue	L'enfant prodigue	Drama	1907	90	5.7000

5 rows × 22 columns

Vérification des données

- S'assurer qu'aucune donnée ne manque

Export de `df_info`

- Finalisation et Export du df

In [30]:
```
# Export final avec compression GZIP
df_merge_2.to_csv('../../donnees/data/df_info.csv.gz',
                  index=False,
                  encoding='utf-8',
                  compression='gzip')
```