Rapport de projet – Analyse de données et apprentissage automatique

Objectif général

Ce projet a pour objectif de conduire un pipeline complet d'analyse de données, depuis le choix du dataset jusqu'à l'application d'un ou plusieurs modèles de machine learning.
Vous travaillerez en groupe (3 à 4 étudiants) et documenterez toutes les étapes de votre démarche dans ce rapport.
Chaque section doit être remplie de manière rigoureuse, avec des explications claires et des justifications.
Le rapport servira à l'évaluation finale du module pour 40% de la note finale.

Partie 1 – Définition du sujet et choix du dataset

1.1 Thématique générale

Décrivez le thème général de votre projet et la problématique que vous souhaitez explorer.

- Quel est le domaine choisi (santé, culture, sport, environnement, économie, société, etc.) ?
- Quelle question ou quel objectif souhaitez-vous atteindre à travers l'analyse des données ?
- Pourquoi ce sujet vous intéresse-t-il ?
- En quoi ce sujet présente-t-il un intérêt pour la société, pour un secteur ou pour une entreprise ?

Réponse :

- **Domaine choisi : Culture du Cinéma** – Analyse prédictive des succès cinématographiques
- **Problématique :** Nous souhaitons explorer les facteurs associés au succès des films et préparer un jeu de données exploitable pour prédire un indicateur de performance à partir de variables explicatives.
- **Objectif :** construire un dataset propre et riche à partir de l'API TMDB, réaliser une exploration descriptive pour expliquer/prédire la performance**.**
- **Intérêt personnel :** Ce sujet nous intéresse car il combine notre passion pour le cinéma avec l'analyse de données.

- **Intérêt sociétal/économique :**
  - **Pour l'industrie cinématographique** : Optimiser les investissements et réduire les risques financiers
  - **Pour les plateformes de streaming** : Améliorer les algorithmes de recommandation
  - **Pour la recherche académique** : Comprendre les tendances culturelles et l'évolution du cinéma

1.2 Recherche et sélection du dataset

Recherchez un dataset sur une plateforme ouverte (sources possibles : Kaggle, UCI, data.gouv.fr,INSEE,WorldBank,etc.).
Vérifiez qu'il répond aux critères de qualité et de pertinence énoncés ci-dessous.

Informations générales sur le dataset :

- ***Nom du dataset :*** Films TMDB (The Movie Database)
- ***Source et lien d'accès :*** API TMDB : https://www.themoviedb.org/
- ***Auteur ou organisation : The*** Movie Database (TMDB) - Communauté open-source
- ***Taille (nombre de lignes et de colonnes)*** : 2,521 films (plus l'en-tête = 2,522 lignes) / 14 variables enrichies
- ***Format du fichier (CSV, JSON, Excel, etc.) :*** CSV avec encodage UTF-8

Vérification de la qualité :

- ***Le dataset est-il récent ?*** Oui, TMDB est continuellement mis à jour.
- ***Les variables sont-elles clairement nommées et documentées ?*** Oui.

  (id, title, original_language, release_date, vote_average, vote_count, polularity, genre_ids, production_countries). Les variables sont bien documentées.

- ***Contient-il suffisamment de données (au moins plusieurs centaines de lignes) ?*** Oui. L'API permet de parcourir des pages de résultats (environ 2500 lignes)
- ***Le dataset comporte-t-il une variable cible que vous pourrez prédire ou expliquer ?*** Oui,
  - Variable principale : popularity (prédiction du succès)
  - Variable alternative : vote_average (prédiction de la qualité)

- ***Les données semblent-elles complètes et cohérentes ?*** Globalement bonne avec 2500+ films avec données détaillées mais avec certaines valeurs peuvent manquer donc un nettoyage sera prévu.

Justification du choix :

Expliquez en quelques phrases pourquoi vous avez choisi ce dataset plutôt qu'un autre :

- En quoi est-il adapté à votre question de recherche ?
- Quels sont ses avantages ?
- Quelles sont ses limites ou difficultés potentielles (taille, biais, manque de variables, etc.) ?

Réponse :

- En quoi est-il adapté à notre question de recherche ?
  - Le dataset contient des métriques de succès (popularity, vote_average, vote_count)
  - Il inclut des caractéristiques descriptives (genres, langue, année de sortie)
  - Il permet d'analyser les facteurs influençant le succès d'un film

- Avantages :
  - Données récentes : API mise à jour en temps réel
  - Données complètes : Métadonnées riches (genres, dates, notes, etc.)
  - Accessibilité : API gratuite et bien documentée
  - Taille adaptée : ~2500 films (suffisant pour l'analyse, pas trop volumineux)
  - Variables cibles claires : popularity et vote_average sont des métriques quantifiables

- Limites ou difficultés potentielles :
  - Valeurs manquantes : Certaines colonnes (overview, backdrop_path) peuvent avoir des valeurs manquantes
  - Biais de sélection : Les films dans le dataset sont triés par popularité (peut créer un biais)
  - Langue: Les titres sont localisés (peut nécessiter un traitement spécial)

1.3 Validation du dataset

Complétez la grille suivante avant de commencer votre analyse :

Critère	Question	Réponse (Oui/Non)	Détail/Justification
Pertinence	Le dataset permet-il de répondre à votre question de départ ?	Oui	Indicateurs de performance (popularité, notes) + métadonnées explicatives.
Clarté	Les variables sont-elles bien nommées et compréhensibles ?	Oui	Documentation TMDB claire, champs standards (title, vote_average, genres, etc.).

Critère	Question	Réponse (Oui/Non)	Détail/Justification
Propreté	Les données semblent-elles utilisables sans nettoyage majeur ?	Plutôt oui	Quelques champs manquants ou multi-valués à normaliser, nettoyage prévu mais faisable.
Taille	Le dataset est-il d'une taille adaptée à votre analyse ?	Oui	Extraction paginée permettant plusieurs milliers de films.
Accessibilité	Le format est-il compatible avec Python (CSV, XLSX) ?	Oui	API JSON -> transformation vers CSV avec pandas.
Actualité	Les données sont-elles récentes ou encore valides ?	Oui	Base vivante, mise à jour en continue, extraction datée du projet.

Partie 2 – Exploration initiale des données

2.1 Chargement et aperçu du dataset

Importez le dataset dans un notebook Python à l'aide de pandas.

Questions :

- Combien de lignes et de colonnes comporte votre dataset ?
- Quelles sont les principales variables (nom et type) ?
- Quelle est la signification de chacune ?
- Identifiez-vous des valeurs manquantes ou des incohérences ?

Réponse *(à rédiger et JUSTIFIER avec des extraits ou résultats du code) :*

- ***Question 1 :*** Combien de lignes et de colonnes comporte votre dataset ?

Le dataset comporte 2500+ lignes et 14 colonnes. Ce résultat a été obtenu en chargeant le fichier CSV avec pandas et en utilisant la méthode `df.shape`, qui retourne un tuple (nombre_lignes, nombre_colonnes). Cela correspond à notre extraction de 125 pages de résultats TMDB à raison de 20 films par page.

```
[1]                                                                                                                    Python
...  Fichier chargé: C:\Users\flori\OneDrive\Bureau\TMDB\data/processed/films_tmdb.csv
     Shape: (2500, 14)
```

	id	title	original_title	original_language	release_date	adult	popularity	vote_average	vote_count	genre_ids	overview	original_country	
0	1413602	WWE Raw on Netflix Premier Post-Show	WWE Raw on Netflix Premier Post-Show	en	2025-01-06	False	447.0546	6.500	1	NaN	NaN	NaN	/xYl50zKRbX8pSu

Extrait du code utilisé :

```
df = pd.read_csv(CSV_PATH)
print("Shape:", df.shape)
df.head(3)
```

- ***Question 2 :*** Quelles sont les principales variables (nom et type) ?

Le dataset contient 14 variables principales réparties comme suit :

- Variables quantitatives continues (3) : `popularity` (float64), `vote_average` (float64), `original_country` (float64 avec valeurs manquantes)
- Variables quantitatives discrètes (2) : `id` (int64), `vote_count` (int64)
- Variables qualitatives nominales (7) : `title`, `original_title`, `original_language`, `genre_ids`, `backdrop_path`, `poster_path`, `adult` (bool)
- Variables temporelles (1) : `release_date` (string, format YYYY-MM-DD)
- Variables texte libre (1) : `overview` (résumé du film)

Cette classification a été obtenue en analysant les types de données retournés par `df.dtypes` et en les catégorisant selon leur nature statistique.

```
[2]
...
Dtypes:
 id                   int64
title                object
original_title       object
original_language    object
release_date         object
adult                  bool
popularity          float64
vote_average        float64
vote_count            int64
genre_ids            object
overview             object
original_country    float64
backdrop_path        object
poster_path          object
dtype: object

Résumé numérique (quantitatives):
                count          mean           std      min          25%         50%          75%          max
id             2500.0  432901.255600  476917.398727  11.0000  10544.500000  244069.0000  804540.250000  1.566841e+06
popularity     2500.0      15.412321      22.908159   5.5294      8.614075      11.2588      13.890325  4.470546e+02
vote_average   2500.0       6.575097       1.428909   0.0000      6.187250       6.8000       7.359250  1.000000e+01
vote_count     2500.0    4116.889600    5410.759201   0.0000    124.000000    2061.0000    5952.250000  3.813700e+04
original_country  0.0            NaN            NaN      NaN           NaN          NaN            NaN           NaN
...
0.75   13.890325
0.95   31.269600
0.99   95.867569
1.00  447.054600
```

Extrait de code utilisé :

```python
# 2.1 - Aperçu: types, description et valeurs manquantes
print("\nDtypes:\n", df.dtypes)

print("\nRésumé numérique (quantitatives):")
print(df.describe(include=[np.number]).T)

print("\nAperçu des variables non-numériques:")
print(df.describe(include=[object, 'bool']).T)

missing = df.isna().sum().sort_values(ascending=False)
missing = missing[missing > 0]
print("\nValeurs manquantes par variable (>0):\n", missing)

# Distribution simple pour quelques variables clés
for col in ["vote_average", "vote_count", "popularity"]:
    if col in df.columns:
        q = df[col].quantile([0, .25, .5, .75, .95, .99, 1.0]).to_frame(name=col)
        print(f"\nQuantiles {col}:\n", q)
```

- ***Question 3 :*** Quelle est la signification de chacune ?

Justifications détaillées :

id : id unique TMDB du film (entier)

title : titre du film localisé en français (fr)

original_title : titre original du film dans sa langue d'origine

original_language : langue originale (ex: "en", "fr", "es")

release_date : date de sortie du film au format YYYY-MM-DD

adult : indicateur booléen : `True` si le film est destiné aux adultes, `False` sinon

popularity : score de popularité TMDB calculé à partir de plusieurs facteurs (consultations, votes, etc.). Valeur continue généralement entre 0 et plusieurs centaines

vote_average : note moyenne des utilisateurs TMDB sur 10

vote_count : nb total de votes reçus par le film

genre_ids : liste d'identifiants de genres TMDB séparés par le caractère "|" (ex: "28|12|878" = Action, Aventure, Science-fiction). Format chaîne de caractères

overview : résumé du film en français

backdrop_path : Chemin relatif vers l'image de fond du film (nettoyé)

poster_path : Chemin relatif vers l'affiche du film (nettoyé)

- ***Question 4 :*** Identifiez-vous des valeurs manquantes ou des incohérences ?

Oui, plusieurs variables contiennent des valeurs manquantes, et quelques incohérences mineures ont été détectées.

```python
    # 2.1 - Incohérences simples et corrections de types
    # release_date -> datetime et release_year
    if "release_date" in df.columns:
        df["release_date_parsed"] = pd.to_datetime(df["release_date"], errors="coerce")
        df["release_year"] = df["release_date_parsed"].dt.year
        invalid_dates = df["release_date"].isna().sum() + df["release_date_parsed"].isna().sum()
        print("Lignes avec date invalide (brutes + parse):", int(invalid_dates))

    # cast types usuels
    for col, to_type in [("adult", "boolean"), ("vote_count", "Int64"), ("vote_average", "float"), ("popularity", "float")]:
        if col in df.columns:
            try:
                if to_type == "boolean":
                    df[col] = df[col].astype("boolean")
                else:
                    df[col] = df[col].astype(to_type)
            except Exception as e:
                print(f"Type cast échoué pour {col} -> {to_type} : {e}")

    # doublons sur id
    if "id" in df.columns:
        before = len(df)
        df = df.drop_duplicates(subset=["id"])  # on ne réécrit pas le CSV ici; juste le comptage
        print("Doublons supprimés (id):", before - len(df))

    # vérifier valeurs négatives aberrantes
    for col in ["vote_count", "vote_average", "popularity"]:
        if col in df.columns:
            neg = (df[col] < 0).sum()
            if neg > 0:
                print(f"Avertissement: {neg} valeurs négatives dans {col}")
```

Python

```
Lignes avec date invalide (brutes + parse): 6
Doublons supprimés (id): 249
```

Résultat obtenu :

```
Colonnes avec valeurs manquantes:
         Colonne  Valeurs_manquantes  Proportion_%
original_country                2500        100.00
        overview                 297         11.88
   backdrop_path                  63          2.52
       genre_ids                  17          0.68
     poster_path                  12          0.48
    release_date                   3          0.12
```

Analyse des valeurs manquantes :

- `original_country`: toutes les valeurs sont manquantes. Cette variable n'a pas été correctement extraite de l'API TMDB. (ps : on a fini par supprimer cette colonne)
- `overview`: + de 10% des films n'ont pas de résumé. On va surement conserver ces lignes mais gérer les NaN lors de l'analyse textuelle
- `backdrop_path`: certains films n'ont pas d'image de fond mais ce n'ai pas très grave pour notre analyse
- `genre_ids` : très peu de films sans genre
- `poster_path`: impact négligeable.
- `release_date` : très peu de dates manquantes.

2.2 Typologie des données

Classez vos variables selon leur type :

- Variables quantitatives continues (ex. revenu, âge)
- Variables quantitatives discrètes (ex. nombre d'enfants)
- Variables qualitatives nominales (ex. pays, couleur)
- Variables qualitatives ordinales (ex. niveau d'éducation)
- Variables temporelles (ex. date, année)
- Autres (texte libre, image, etc.)

Questions :

- Quelles sont les variables les plus importantes pour votre analyse ?
- Y a-t-il une variable cible que vous cherchez à prédire ou expliquer ?

Classification des variables selon leur type

Justification avec code :

```
quant_cont = []
quant_disc = []
qual_nom = []
qual_ord = []
temporelles = []
autres = []

for c in df.columns:
    dt = df[c].dtype
    if c in {"release_date", "release_date_parsed", "release_year"}:
        temporelles.append(c)
    elif pd.api.types.is_float_dtype(dt):
        quant_cont.append(c)
    elif pd.api.types.is_integer_dtype(dt):
        # vote_count est un bon exemple de discrète
        quant_disc.append(c)
    elif pd.api.types.is_bool_dtype(dt) or str(dt) == "boolean":
        qual_nom.append(c)
    else:
        # objets textuels, catégories non ordonnées
        if c in {"title", "original_title", "overview"}:
            autres.append(c)
        else:
            qual_nom.append(c)

print("Quantitatives continues:", quant_cont)
print("Quantitatives discrètes:", quant_disc)
print("Qualitatives nominales:", qual_nom)
print("Qualitatives ordinales:", qual_ord)  # généralement aucune ici
print("Temporelles:", temporelles)
print("Autres (texte):", autres)
```

Résultat obtenu :

```
Quantitatives continues: ['popularity', 'vote_average', 'original_country']
Quantitatives discrètes: ['id', 'vote_count']
Qualitatives nominales: ['original_language', 'adult', 'genre_ids', 'backdrop_path', 'poster_path']
Qualitatives ordinales: []
Temporelles: ['release_date', 'release_date_parsed', 'release_year']
Autres (texte): ['title', 'original_title', 'overview']
```

Détail de chaque catégorie

1. Variables quantitatives continues

Variables : `popularity`, `vote_average`, `original_country`

Justification : Ces variables sont de type `float64` et peuvent prendre n'importe quelle valeur réelle dans un intervalle. (`popularity`: Score continu variant de 5.53 à 447.05, moyenne = 15.41, `vote_average` : Note moyenne continue de 0.0 à 10.0, moyenne = 6.58, `original_country`: Variable continue mais entièrement manquante (à exclure))

```
Résumé numérique (quantitatives):
                  count           mean            std      min          25%          50%          75%           max
id               2500.0  432901.255600  476917.398727  11.0000  10544.500000  244069.0000  804540.250000  1.566841e+06
popularity       2500.0      15.412321      22.908159   5.5294      8.614075      11.2588      13.890325  4.470546e+02
vote_average     2500.0       6.575097       1.428909   0.0000      6.187250       6.8000       7.359250  1.000000e+01
vote_count       2500.0    4116.889600    5410.759201   0.0000    124.000000    2061.0000    5952.250000  3.813700e+04
original_country    0.0            NaN            NaN      NaN           NaN          NaN           NaN           NaN
...
```

2. Variables quantitatives discrètes

Variables : `id`, `vote_count`

Justification : Ces variables sont de type `int64` et ne peuvent prendre que des valeurs entières. (`id`: Identifiant unique, valeurs entières de 11 à 1,566,841, `vote_count`: Nombre de votes (entier ≥ 0), variant de 0 à 38,137 votes, médiane = 2,061)

Pour identifier les variables discrètes :On se base sur le type de données(`int64` pour `id` et `vote_count`) et le fait qu'elles ne peuvent prendre que des valeurs entières, même si elles apparaissent dans le même `describe()` que les variables continues.

Statistiques des variables discrètes (extrait du describe ci-dessus) :

- `id`: min=11, max=1,566,841, médiane=244,069

- `vote_count`: min=0, max=38,137, médiane=2,061

3. Variables qualitatives nominales

Variables : `original_language`, `adult`, `genre_ids`, `backdrop_path`, `poster_path`

Justification: Variables catégorielles sans ordre naturel, chaque modalité est distincte. (`original_language`: 38 langues différentes (code ISO), majorité = "en" (1969 films), `adult`: Variable binaire (False/True), ici tous False (2500), `genre_ids`: Chaîne de caractères avec IDs séparés par "|", 832 combinaisons uniques,

```
================================================================================
STATISTIQUES DES VARIABLES QUALITATIVES ET TEXTUELLES
================================================================================

Aperçu des variables non-numériques:
                  count unique                                                top  freq
title              2500   2221                                          Maléfique     4
original_title     2500   2211                                          Halloween     4
original_language  2500     38                                                 en  1969
release_date       2497   1852                                         2025-09-19    12
adult              2500      1                                              False  2500
genre_ids          2483    832                                                 18   120
overview           2203   1985  Maléfique est une belle jeune femme au cœur pu...     4
backdrop_path      2437   2196                    /4hfcpHmMEgmFTdnVx4XCtM6dgCG.jpg     4
poster_path        2488   2240                    /v1ggksl61yDoTL3WqqUyoKhN6TH.jpg     4

================================================================================
```

```
Problems  Output  Debug Console  Terminal  Ports
PS C:\Users\flori\OneDrive\Bureau\TMDB> python -c "import pandas as pd; import sys; sys.path.append('.'); from config import
 OUTPUT_CSV; df = pd.read_csv(OUTPUT_CSV); print('Top 5 langues:'); print(df['original_language'].value_counts().head())"
Top 5 langues:
original_language
en    1969
ja     100
ko      61
es      54
fr      49
Name: count, dtype: int64
PS C:\Users\flori\OneDrive\Bureau\TMDB>
```

4. Variables qualitatives ordinales

Variables : Aucune

5. Variables temporelles

Variables : `release_date`, `release_date_parsed`, `release_year`

Justification : Variables représentant des dates/heures. (`release_date`: Format string "YYYY-MM-DD" (ex: "2025-01-06"), `release_date_parsed`: Conversion en `datetime64[ns]` pour analyse temporelle, `release_year`: Extraction de l'année (float64 après parsing))

Extrait de code :

```
# release_date -> datetime et release_year
if "release_date" in df.columns:
    df["release_date_parsed"] = pd.to_datetime(df["release_date"], errors="coerce")
    df["release_year"] = df["release_date_parsed"].dt.year
    invalid_dates = df["release_date"].isna().sum() + df["release_date_parsed"].isna().sum()
    print("Lignes avec date invalide (brutes + parse):", int(invalid_dates))
```

Note :`release_date_parsed` et `release_year` sont créées lors de l'exploration (cellule 2), pas dans le CSV original. Cela explique pourquoi ces colonnes apparaissent dans la classification des variables temporelles du notebook.

6. Autres (texte libre)

Variables :`title`, `original_title`, `overview`

Justification : Variables textuelles non structurées, utilisables pour l'analyse de texte (NLP). (`title`: Titre localisé en français, 2221 titres uniques, `original_title`: Titre original, 2211 titres uniques, `overview`: Résumé/synopsis en français (texte libre), 1985 résumés uniques, 297 manquants

Justification avec statistiques de `df.describe()` : Les statistiques pour les variables texte sont extraites du résultat de `df.describe(include=[object, 'bool'])` exécuté dans le notebook (cellule 1).

	count	unique	top	freq
title	2500	2221	Maléfique	4
original_title	2500	2211	Halloween	4
original_language	2500	38	en	1969
release_date	2497	1852	2025-09-19	12
adult	2500	1	False	2500

- **Question 1 : Quelles sont les variables les plus importantes pour votre analyse ?**

Les variables les plus importantes pour analyser et prédire le succès des films sont:

- Variables cibles (à prédire) : `popularity` : Indicateur principal de succès (métrique composite TMDB) / `vote_average`: Qualité perçue par les utilisateurs (0-10)
- Variables explicatives principales :`vote_count`: Nombre de votes (corrélé à la visibilité/notoriété) / `genre_ids`: Genres du film (facteur déterminant du succès) /

`release_date`,`release_year`: Période de sortie (effet temporal) /`original_language`: Langue originale (impact sur l'audience internationale) /`overview`: Résumé (pour analyse textuelle et sentiment)

- Variables secondaires :`title`, `original_title` : Pour analyse de titre (impact marketing) / `adult` : Filtrage (ici tous False, peu informatif)
- Variables à exclure : `original_country` : 100% manquantes / `id`, `backdrop_path`, `poster_path` : soient elles sont absente soient pas très utiles pour la prédiction.

```
Variables numériques clés :
        popularity  vote_average    vote_count
count  2500.000000   2500.000000   2500.000000
mean     15.412321      6.575097   4116.889600
std      22.908159      1.428909   5410.759201
min       5.529400      0.000000      0.000000
25%       8.614075      6.187250    124.000000
50%      11.258800      6.800000   2061.000000
75%      13.890325      7.359250   5952.250000
max     447.054600     10.000000  38137.000000
```

```
1. popularity:
  Médiane = 11.26
  Max = 447.05
  Écart-type = 22.91
  -> Grande variabilité (std = 22.91) -> bonne variable cible

2. vote_average:
  Médiane = 6.80/10
  Écart-type = 1.43
  -> Distribution concentrée (std = 1.43)
  -> Corrélée avec le succès commercial
ion attendue avec popularity
  -> Indicateur de notoriété
```

```
4. genre_ids:
  832 combinaisons uniques
  Genre le plus fréquent : "18" apparaît dans 861 films
  -> Impact significatif sur le type d'audience
```

Question 2 : Y a-t-il une variable cible que vous cherchez à prédire ou expliquer ?

Oui, deux variables cibles potentielles :

1. `popularity` : Variable cible principale

- Métrique composite reflétant le succès global
- Type : Variable quantitative continue

2. `vote_average`: Variable cible alternative

- Qualité perçue par les utilisateurs
- Type : Variable quantitative continue (0-10)

- Justification du choix de la variable cible principale

Raison : Pertinence métier

`popularity` combine plusieurs facteurs (consultations, votes, tendances). Indicateur synthétique du succès commercial et médiatique. Plus représentatif du "succès" qu'une simple note

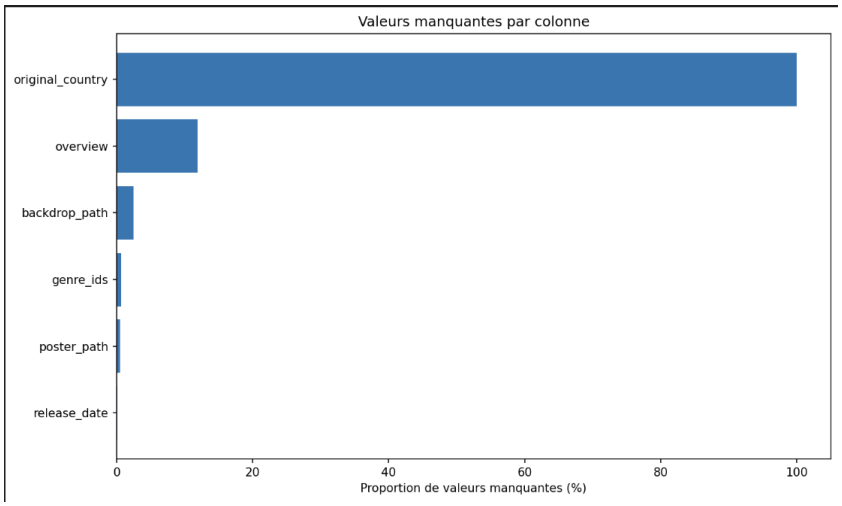
Partie 3 – Nettoyage et préparation du dataset

3.1 Gestion des valeurs manquantes

Questions :

- Quelles colonnes contiennent des valeurs manquantes ?
- Quelle proportion de données manquantes par colonne ?
- Quelle stratégie appliquez-vous ?
- Pourquoi ce choix ?

Réponse *(à rédiger et JUSTIFIER avec des extraits ou résultats du code) :*




```
Colonnes avec valeurs manquantes:
          Colonne  Valeurs_manquantes  Proportion_%
original_country                2500        100.00
        overview                 297         11.88
   backdrop_path                  63          2.52
       genre_ids                  17          0.68
     poster_path                  12          0.48
    release_date                   3          0.12
```

Stratégie appliquée :

1. Suppression d'original_country (100% manquantes) -> colonne non informative.

2. Conservation des autres colonnes avec valeurs manquantes -> proportions faibles (< 12%).

3.2 Détection et traitement des doublons

Questions :

- Avez-vous trouvé des doublons ?
- Comment les avez-vous traités ?

Réponse *(à rédiger et JUSTIFIER avec des extraits ou résultats du code) :*

```
Doublons complets (toutes colonnes identiques): 235
Doublons sur l'ID (identifiant unique): 249

Exemples de doublons sur l'ID:
  IDs en doublon: [1502943 1277988 1242898 1235746 1252309 1256208 1251717 1382406     585
     411]

  Films avec ID 1502943:
       id               title release_date
1502943 Night of the Reaper   2025-10-16
1502943 Night of the Reaper   2025-10-16

  Films avec ID 1277988:
       id    title release_date
1277988 Caramelo   2025-10-07
1277988 Caramelo   2025-10-07

  Films avec ID 1242898:
       id              title release_date
1242898 Predator: Badlands   2025-11-05
1242898 Predator: Badlands   2025-11-05

Doublons sur le titre: 279
```

Traitement :

1. Suppression des doublons sur ID (249 lignes) — conservation de la première occurrence.

2. Suppression des doublons complets (235 lignes).

3. Résultat : dataset final de 2251 lignes (au lieu de 2500).

3.3 Détection des valeurs aberrantes

Vous pouvez utiliser des visualisations (graph, boxplots…).

Questions :

- Quelles variables présentent des valeurs extrêmes ?
- Comment expliquez-vous ces valeurs (erreur, mesure rare, cas particulier) ?
- Avez-vous décidé de les conserver, de les corriger ou de les supprimer ? Pourquoi ?

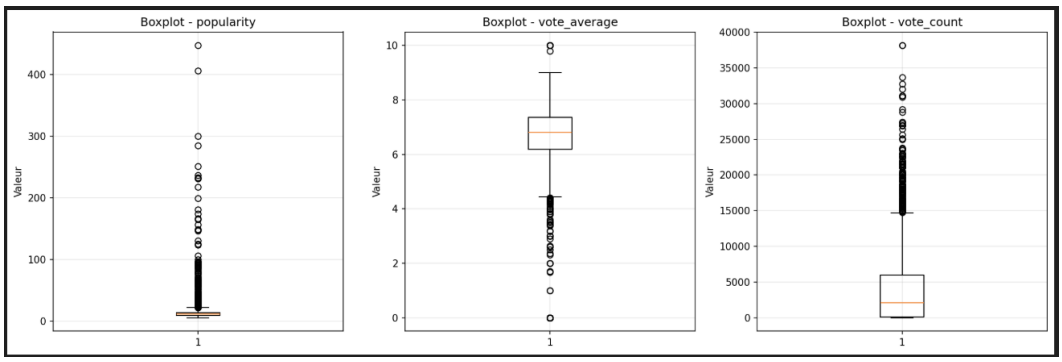
Réponse *(à rédiger et JUSTIFIER avec des extraits ou résultats du code) :*

```
Analyse des valeurs aberrantes (méthode IQR):
--------------------------------------------------------------------------------

popularity:
  Q1: 8.61, Q3: 13.89, IQR: 5.28
  Bornes: [0.70, 21.80]
  Valeurs aberrantes: 223 (8.92%)
  Min réel: 5.53, Max réel: 447.05
  Valeurs extrêmes (haut): [447.0546 406.2517 300.237 ]

vote_average:
  Q1: 6.19, Q3: 7.36, IQR: 1.17
  Bornes: [4.43, 9.12]
  Valeurs aberrantes: 141 (5.64%)
  Min réel: 0.00, Max réel: 10.00
  Valeurs extrêmes (bas): [0. 0. 0.]
  Valeurs extrêmes (haut): [10. 10. 10.]

vote_count:
  Q1: 124.00, Q3: 5952.25, IQR: 5828.25
...
--------------------------------------------------------------------------------
Valeurs zéro ou négatives (potentiellement aberrantes):
  vote_average: 58 valeurs <= 0 (2.32%)
  vote_count: 58 valeurs <= 0 (2.32%)
```



- visualise directement les outliers via les moustaches et points

Explication :

- popularity : films très populaires → valeurs légitimes.
- vote_average : 0.00 (films non notés) et 10.00 (films très bien notés) → valeurs possibles.

- vote_count : films très populaires avec beaucoup de votes → valeurs légitimes.

Décision : conservation des valeurs aberrantes

- Ce sont des valeurs réelles, pas des erreurs.
- Elles représentent des cas particuliers (films très populaires/peu populaires).
- Les supprimer biaiserait l'analyse.

Méthode IQR pour détecter les valeurs aberrantes

3.4 Encodage et mise à l'échelle des variables

Certaines variables doivent être converties en numériques avant d'être utilisées dans un modèle.

Questions :

- Quelles colonnes ont été encodées ?
- Quelle méthode avez-vous utilisée ?
- Pourquoi est-il important de normaliser ou standardiser les données avant l'entraînement des modèles ?

Réponse *(à rédiger et JUSTIFIER avec des extraits ou résultats du code) :*

```
Colonnes catégorielles identifiées:
  original_language: 38 valeurs uniques
  release_date: 1852 valeurs uniques
  adult: 1 valeurs uniques
    Valeurs: {False: 2500}
  genre_ids: 832 valeurs uniques

Colonnes numériques nécessitant normalisation/standardisation:
  popularity:
    Moyenne: 15.41, Écart-type: 22.91
    Min: 5.53, Max: 447.05
    Étendue: 441.53
  vote_average:
    Moyenne: 6.58, Écart-type: 1.43
    Min: 0.00, Max: 10.00
    Étendue: 10.00
  vote_count:
    Moyenne: 4116.89, Écart-type: 5410.76
    Min: 0.00, Max: 38137.00
    Étendue: 38137.00
```

```
3.4 ENCODAGE/SCALING
--------------------------------------------------------------------------------
Colonnes catégorielles: 4
Colonnes numériques à normaliser: 3
```

Méthode utilisée :

- Encodage : non effectué dans cette partie (préparé pour la modélisation).
- Normalisation/standardisation : identifiée comme nécessaire, non appliquée dans cette partie.

Pourquoi normaliser/standardiser ?

1. Éviter que des variables à grande échelle dominent le modèle.
2. Améliorer la convergence des algorithmes.
3. Permettre une comparaison équitable des coefficients.
4. Nécessaire pour certains algorithmes.

Partie 4 – Analyse exploratoire et visualisations

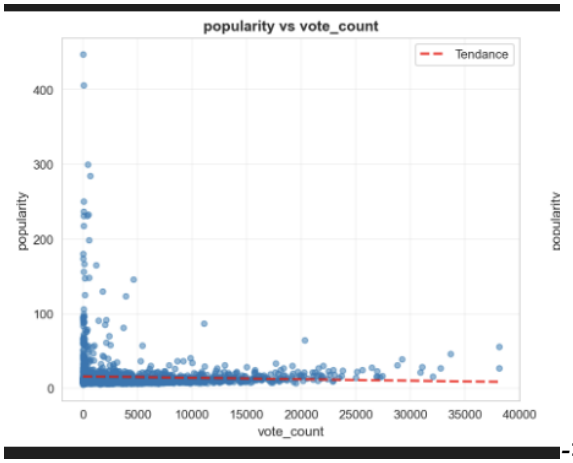
Réalisez une première exploration visuelle pour mieux comprendre les relations entre les variables.

Questions :

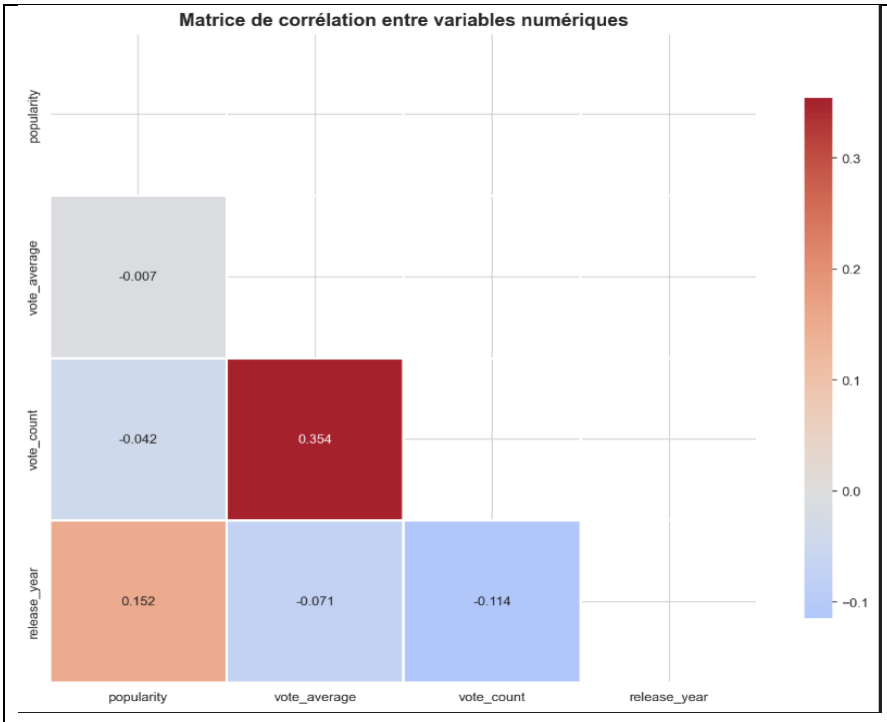
- Quelles relations ou corrélations observez-vous ?
- Quelles variables semblent influencer la variable cible ?
- Quelles hypothèses pouvez-vous formuler ?
- Quelles tendances ou patterns remarquez-vous dans les graphiques ?

Réponse *(à rédiger et JUSTIFIER avec des extraits ou résultats du code) :*

Relations et corrélations observées :



-> *montre une corrélation positive entre vote_count et popularity, mais la relation n'est pas parfaite : on observe des films très populaires avec peu de votes et des films très votés avec popularité modérée, suggérant des cas particuliers (marketing, sorties récentes).*




```
================================================================================
4.1 ANALYSE DES CORRÉLATIONS
================================================================================

Matrice de corrélation (Pearson):
              popularity  vote_average  vote_count  release_year
popularity         1.000        -0.006      -0.037         0.146
vote_average      -0.006         1.000       0.362        -0.068
vote_count        -0.037         0.362       1.000        -0.120
release_year       0.146        -0.068      -0.120         1.000

Corrélations avec 'popularity':
  release_year: 0.146
  vote_average: -0.006
  vote_count: -0.037

Tests de significativité des corrélations (p-value):
  popularity vs vote_average: r=-0.006, p=0.7535
  popularity vs vote_count: r=-0.037, p=0.0629
  popularity vs release_year: r=0.146, p=0.0000 ***
```

- Quelles variables semblent influencer la variable cible ?

```
================================================================================
4.3 VARIABLES INFLUENÇANT LA VARIABLE CIBLE
================================================================================

Variables par ordre d'importance (corrélation absolue):
  release_year: |r|=0.146 (corrélation positive)
  vote_count: |r|=0.037 (corrélation negative)
  vote_average: |r|=0.006 (corrélation negative)

Analyse par groupes de valeurs:

Popularité moyenne par groupe de vote_count:
                    mean  median  count
vote_count_group
Faible (0-100)     21.03   12.52    536
Moyen (100-1K)     17.99   10.93    406
Élevé (1K-10K)     11.99    9.65   1182
Très élevé (>10K)  15.53   13.38    318

Popularité moyenne par groupe de vote_average:
                  mean  median  count
vote_avg_group
<5               21.29   12.59    132
5-6              14.43   10.51    335
6-7              14.91   10.25    971
7-8              15.17   11.45    840
>=8              17.20   12.93    164
```

D'après la matrice de corrélation et les scatter plots(vu avant), vote_count et certaines métriques d'engagement sont les meilleures candidates pour

influencer popularity. Pour vote_average, sa relation avec la cible est moins forte et dépend souvent du vote_count (fiabilité).

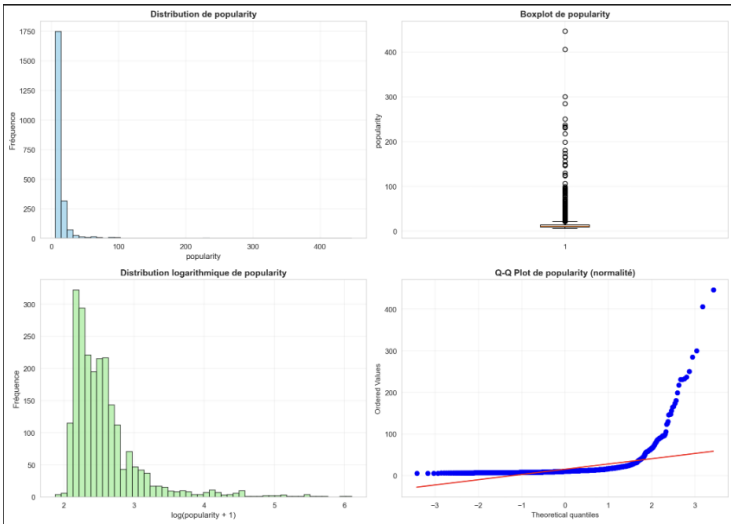
- Quelles hypothèses pouvez-vous formuler ?

Hypothèse 1 : davantage de votes tend à augmenter la visibilité et donc la popularity.

Hypothèse 2 : les notes extrêmes sont moins fiables quand vote_count est faible — elles doivent être considérées séparément.

Hypothèse 3 : la distribution fortement positive de popularity suggère qu'une transformation log peut améliorer la modélisation.

- Quelles tendances ou patterns remarquez-vous dans les graphiques ?



➔ Les histogrammes montrent une forte asymétrie pour popularity : la majorité des films a une faible popularité, une petite fraction concentre des valeurs très élevées. Les boxplots confirment l'existence d'outliers qui correspondent à ces cas réels

```
================================================================================
4.5 TENDANCES ET PATTERNS IDENTIFIÉS
================================================================================

Patterns identifiés:

1. Distribution asymétrique de la popularité
   Description: Asymétrie (skewness) = 9.80
   Interprétation: Distribution fortement asymétrique à droite: peu de films très populaires, beaucoup de films modérément populaires

2. Relation logarithmique entre vote_count et popularity
   Description: Corrélation linéaire: -0.037, Corrélation log: -0.108
   Interprétation: La relation est mieux modélisée avec une transformation logarithmique

3. Deux groupes distincts de popularité
   Description: Groupe élevé (n=1247): moyenne=22.02, Groupe faible (n=1250): moyenne=8.80
   Interprétation: Séparation claire entre films populaires et moins populaires

4. Films récents plus populaires
   Description: Films recents (>=2020): 21.92, Films plus anciens: 11.70
   Interprétation: Les films récents ont tendance à être plus populaires (effet de nouveauté)
```