

TMDB

# *Analyse de Films avec Machine Learning*

*Prédiction de la popularité des films à partir de données TMDB*

**GUILLOU Floriane**

**JARI Jenna**

**KARABAJAKIAN Fred**



**CINEMA**  
PRODUCTION

---

# *Plan de présentation*

1. Contexte et objectifs
2. Choix et description du dataset
3. Nettoyage des données
4. Analyse exploratoire
5. Modélisation prédictive
6. Résultats et conclusion

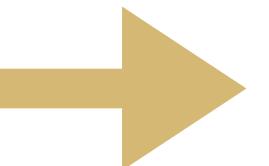
## *Contexte et motivation*

***Quels facteurs influencent le succès d'un film ?***

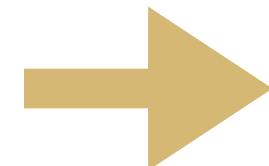


**Domaine**

**Culture et Cinéma**



**Objectif ?**



- **Comprendre les facteurs**
- **Prédire la popularité**
- **Identifier les tendances**

# Intérêt du projet



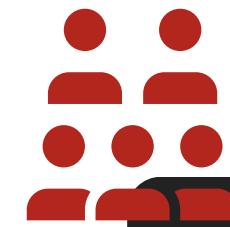
Pour les  
producteurs

Optimiser les décisions  
de production



Pour les  
plateformes

Améliorer les  
recommandations



Pour le public

Mieux comprendre  
les mécanismes de  
popularité

# *Sources des données*



**API TMDB (The Movie Database)**  
<https://www.themoviedb.org/>

**Format : CSV (extrait depuis API JSON)**

**Taille : ~2500 films avec 14 colonnes**

## 2. Choix et description du dataset

# Variables du dataset



### Variables cibles

- **popularity (principale)**
- **vote\_average (alternative)**

### Variables explicatives

- **release\_year**
- **vote\_count**
- **genre\_ids**
- **original\_language**

### Variables descriptives

- **title**
- **overview**

## 2. Choix et description du dataset

# *Qualité du dataset*

### Taille

2500 films  
(suffisant pour l'analyse)

### Complétude

8,26% manquantes  
(taux acceptable)

### Actualité

Données récentes  
(API temps réel)

### Format

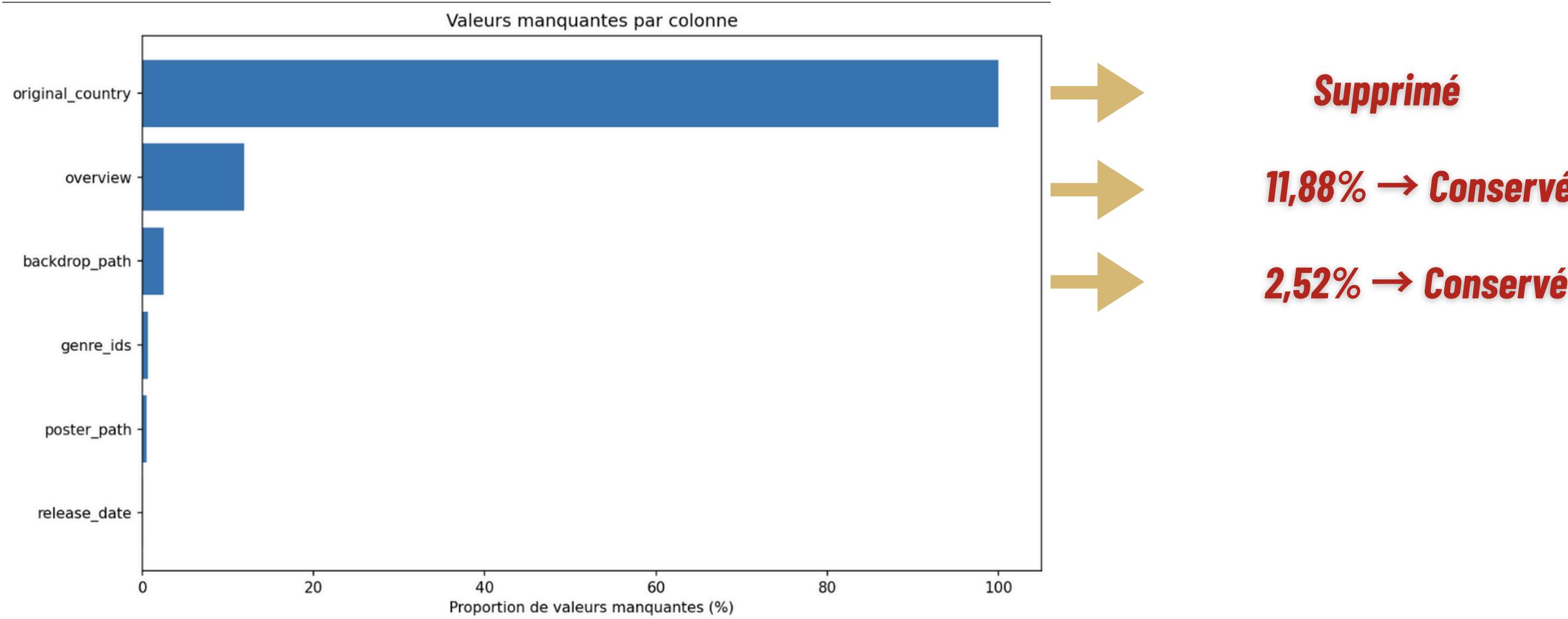
CSV compatible

### Clarté

Variables bien nommées

### 3. Nettoyage des données

## Gestion des valeurs manquantes

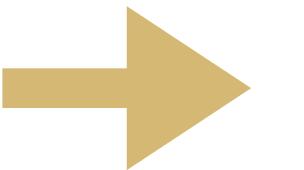


### 3. Nettoyage des données

## *Traitement des doublons*

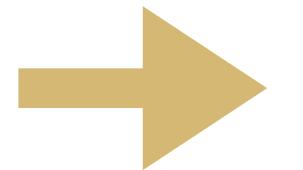
### Doublons détectés

- 249 sur ID
- 235 complets



### Traitement

- Suppression des doublons sur ID
- Suppression des doublons complets



### Résultat

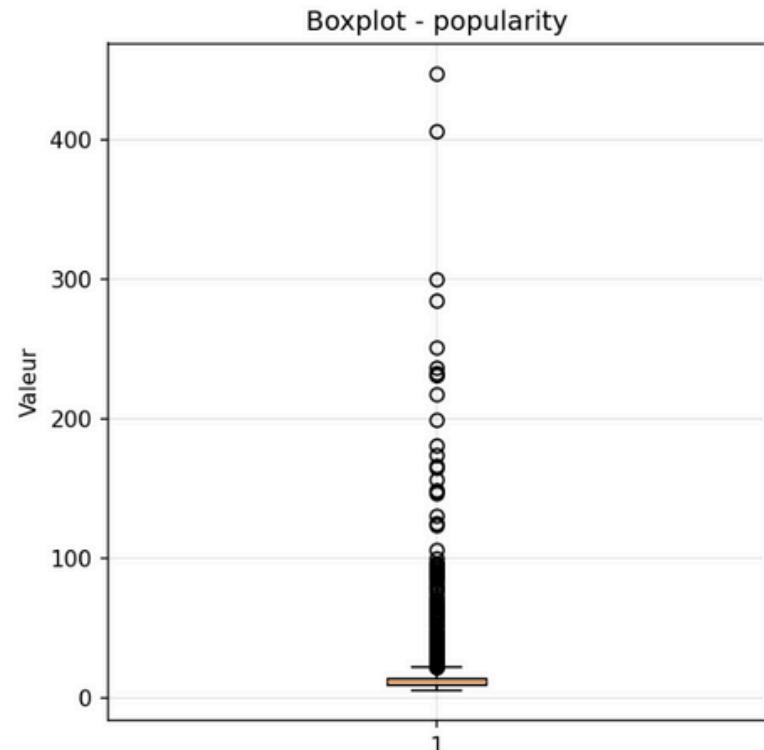
- 2251 films (au lieu de 2500)

### 3. Nettoyage des données

## Valeurs aberrantes

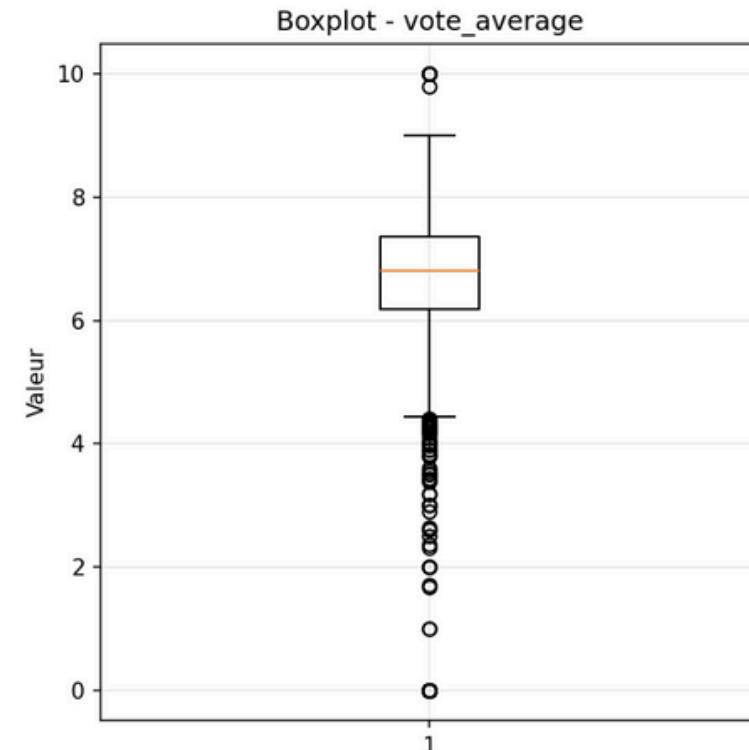
### Popularity

- 223 valeurs (8,92%)
- Max : 44705



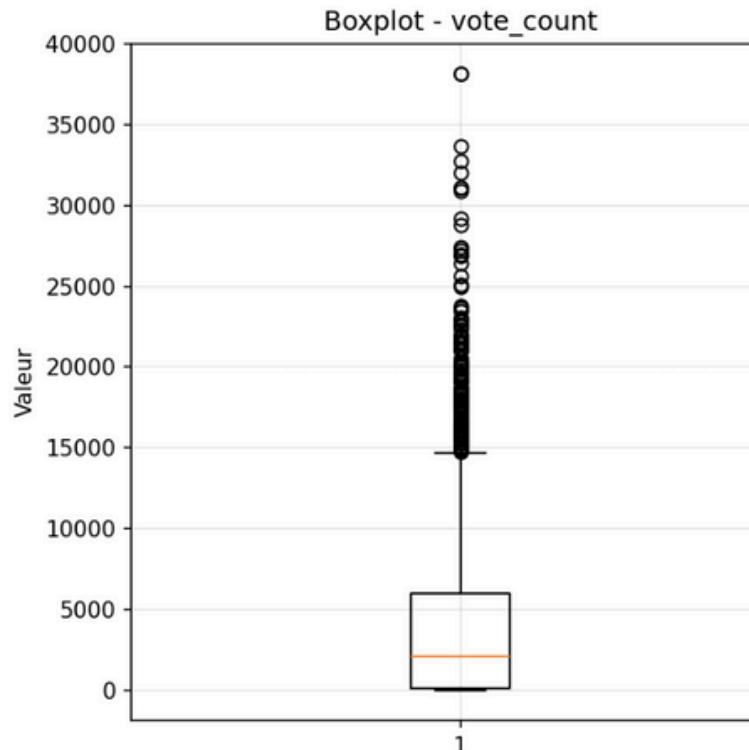
### Vote\_average

- 141 valeurs (5,64%)
- valeurs : 00.0 et 10.00



### Vote\_count

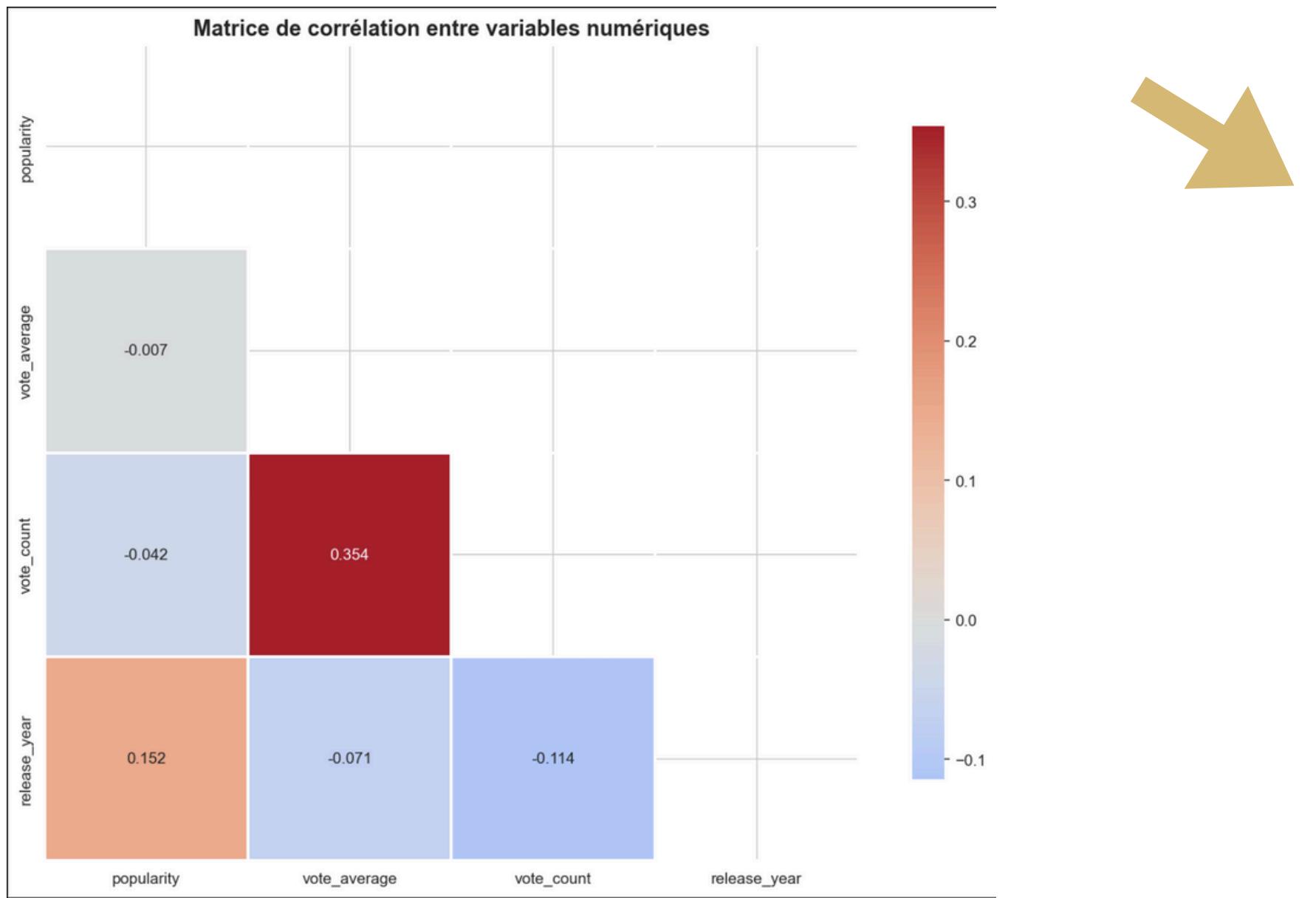
- 146 valeurs (5.84%)
- Max : 38137



Décision : CONSERVÉES  
(Valeurs légitimes)

## 4. Analyse exploratoire

# Visualisation - Matrice de corrélation

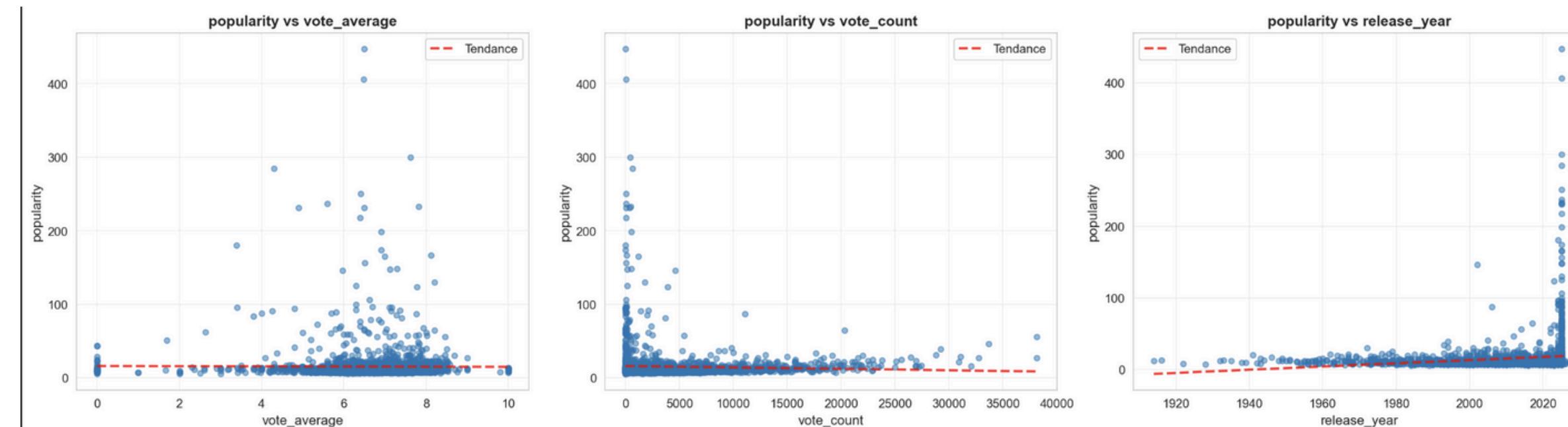


### Observations

- Corrélation modérée entre vote\_average et vote\_count
- Corrélation faible entre popularity et autres

## 4. Analyse exploratoire

# Visualisation - Relations avec la variable cible

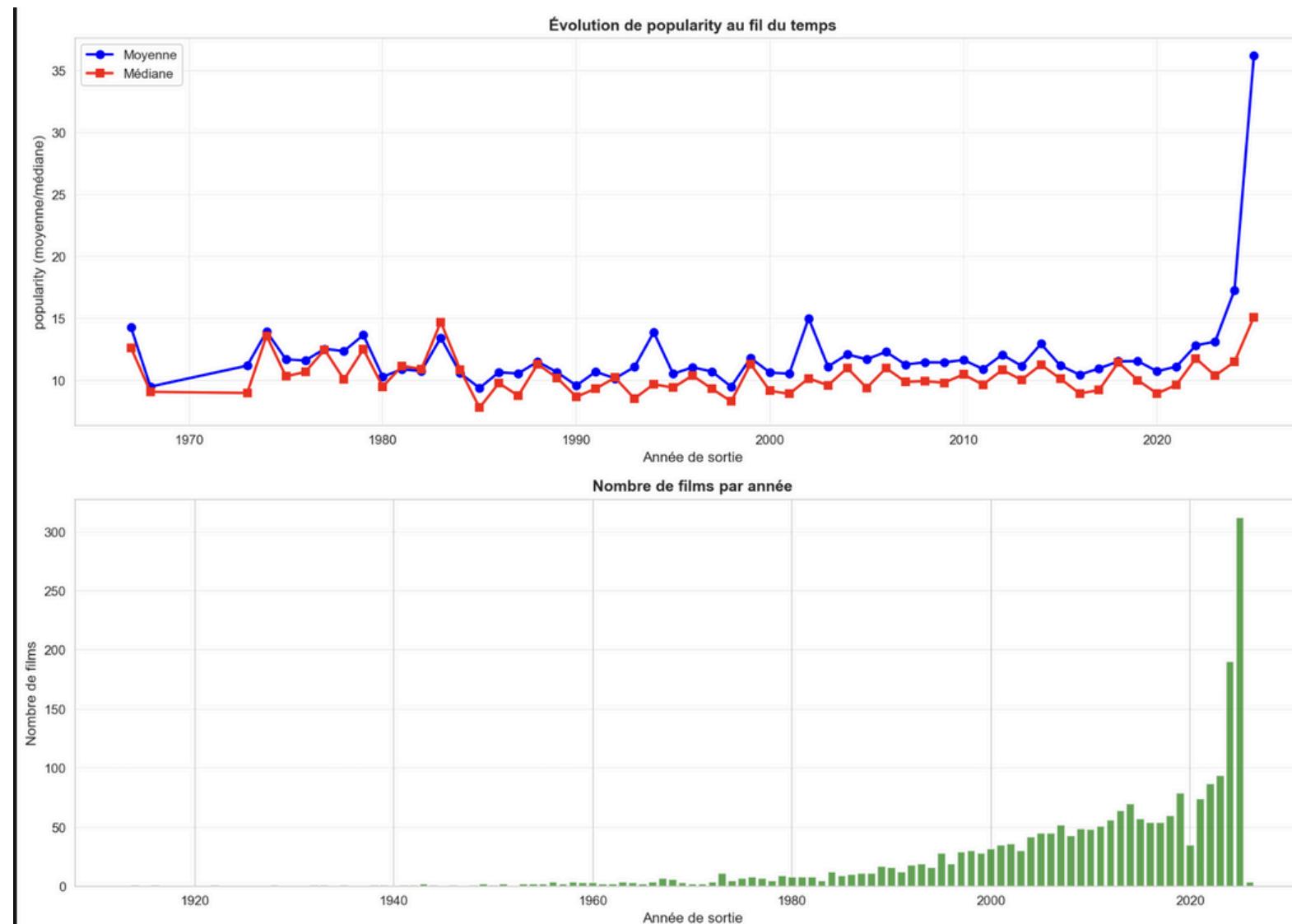


### Observations

- Tendance positive avec release\_year
- Pas de relation claire avec vote\_count ou vote\_average

## 4. Analyse exploratoire

# Visualisation - Analyse temporelle



### Observations

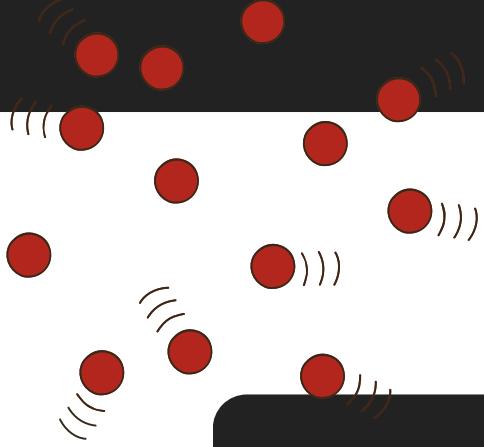
- Augmentation de la popularité au fil du temps
- Pic pour les films récents (2020+)

# *Feature engineering*

### Features créées :

- `log\_vote\_count` : Transformation logarithmique de vote\_count
- `decade` : Décennie de sortie
- `years\_since\_release` : Années depuis la sortie
- `is\_recent` : Film récent ( $\geq 2020$ ) = 1, sinon 0
- Encodage des genres (10 genres les plus fréquents)
- Encodage de `original\_language`
- Normalisation : StandardScaler pour les modèles linéaires et KNN

## 5. Modélisation prédictives



### Modèles linéaires

- Régression Linéaire
- Ridge (régularisation L2)
- Lasso (régularisation L1)

## *Modèles testés*

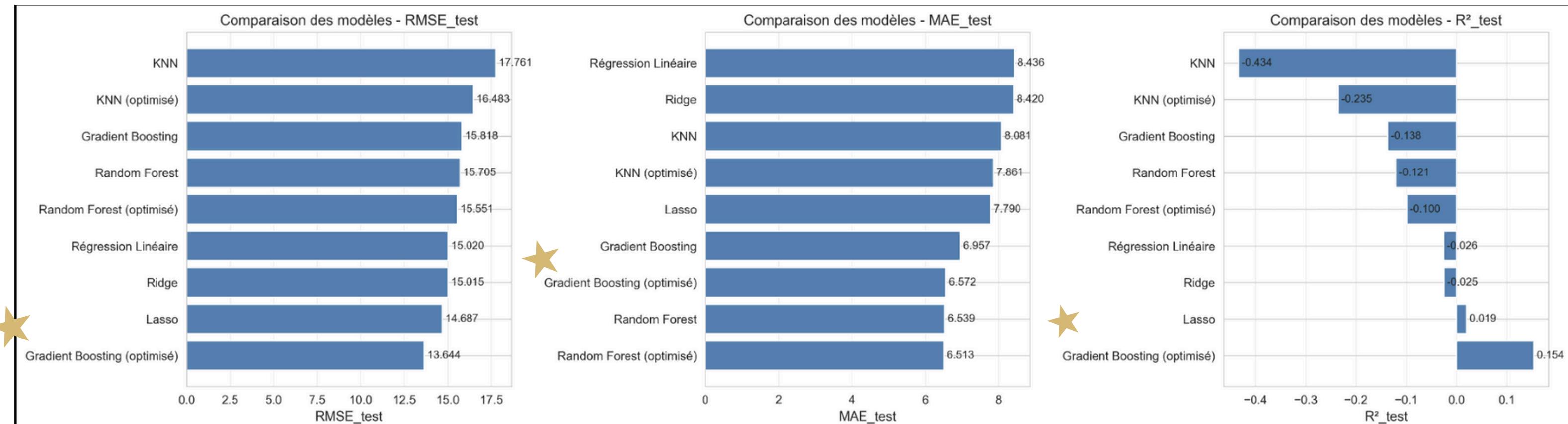
### Modèles non-linéaires

- KNN (K-Nearest Neighbors)
- Random Forest
- Gradient Boosting

(Optimisation : GridSearchCV avec validation croisée (5 folds))

## 5. Modélisation prédictives

# Résultats des modèles



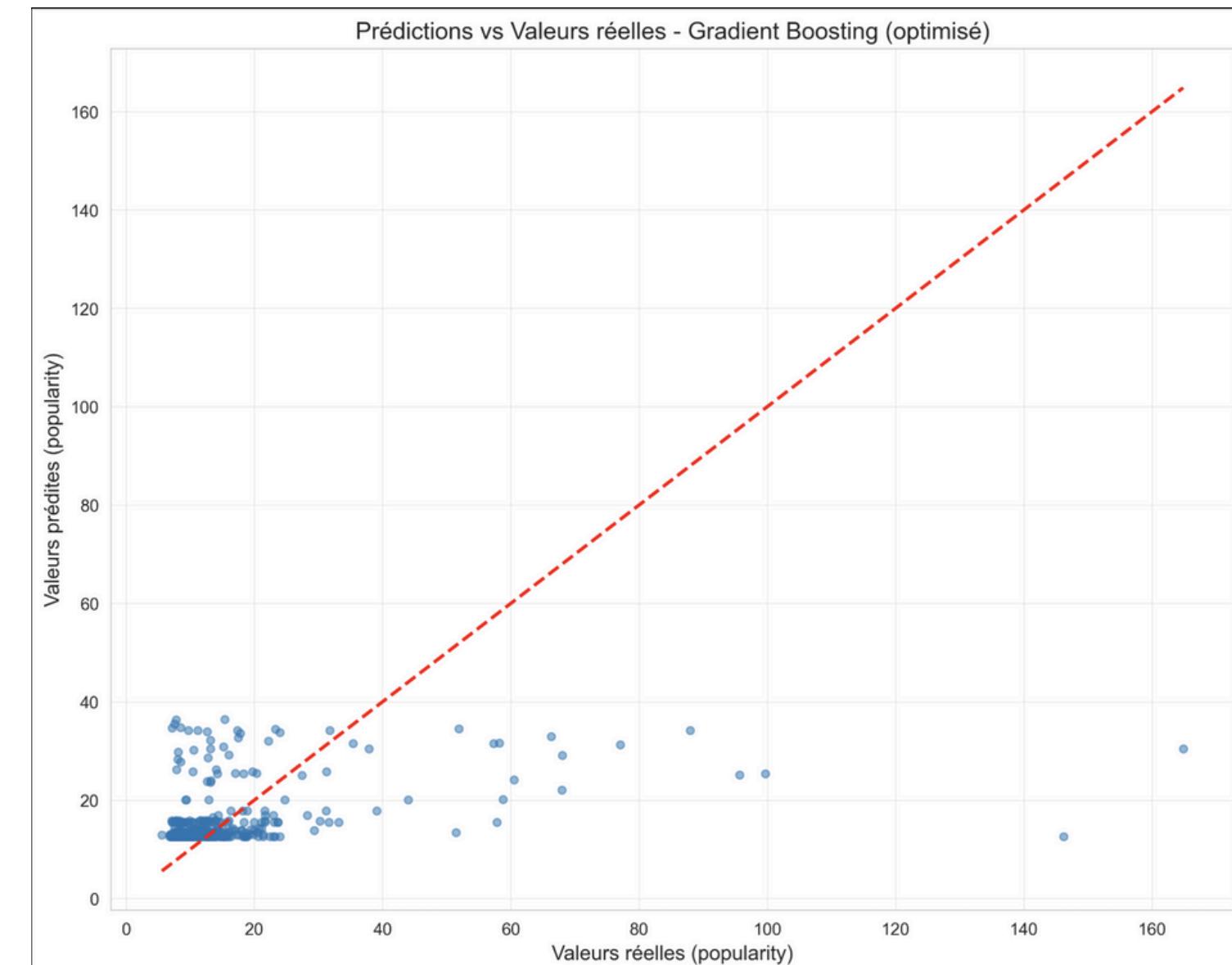
### Observations :

- Les modèles linéaires ont des performances faibles ( $R^2$  négatif)
- L'optimisation améliore significativement les performances

# Visualisation - Prédictions vs Réelles

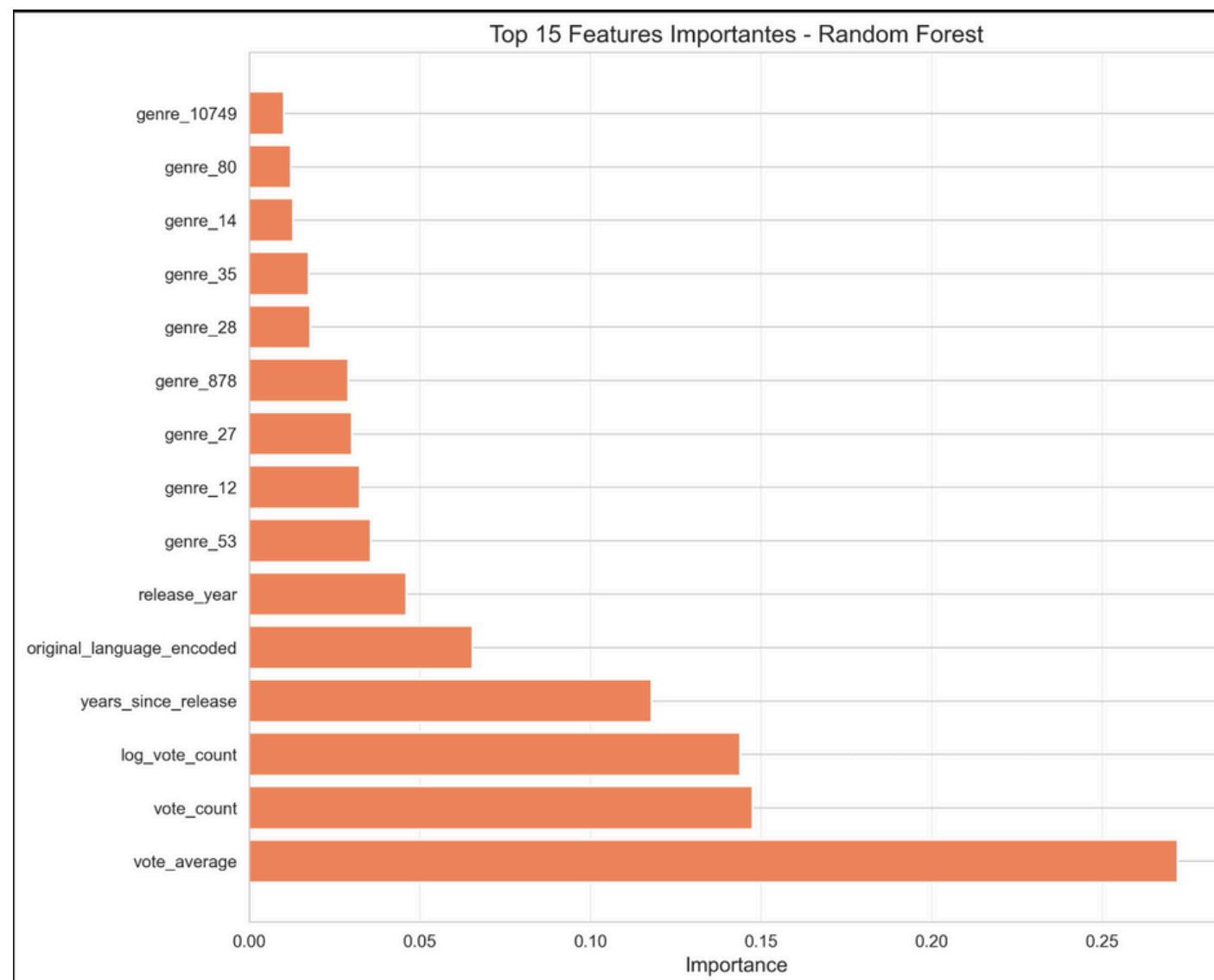
**- Observation :** Les prédictions suivent globalement la tendance

**- Limitations :** Difficulté à prédire les valeurs extrêmes (films très populaires)



## 5. Modélisation prédictives

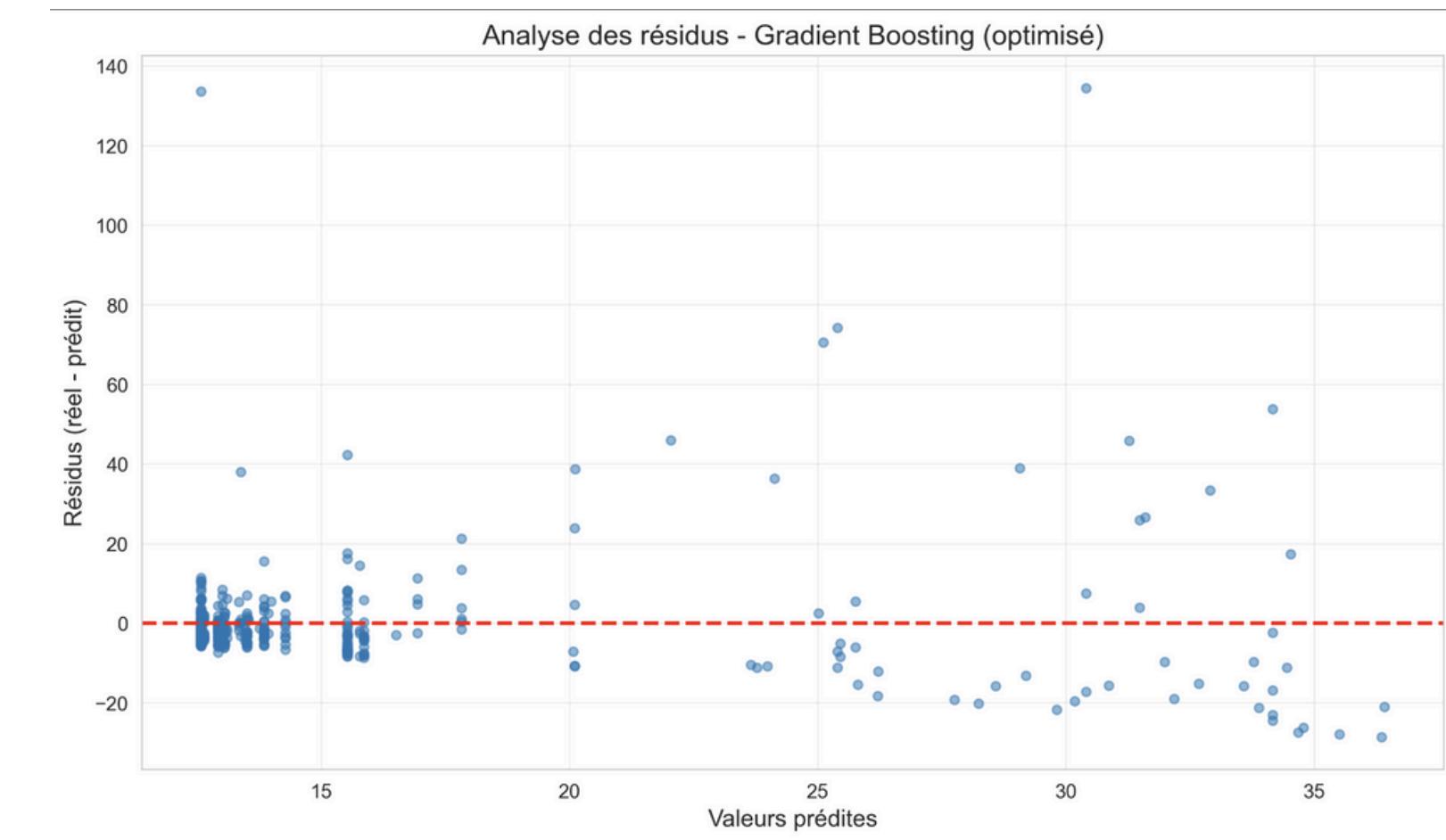
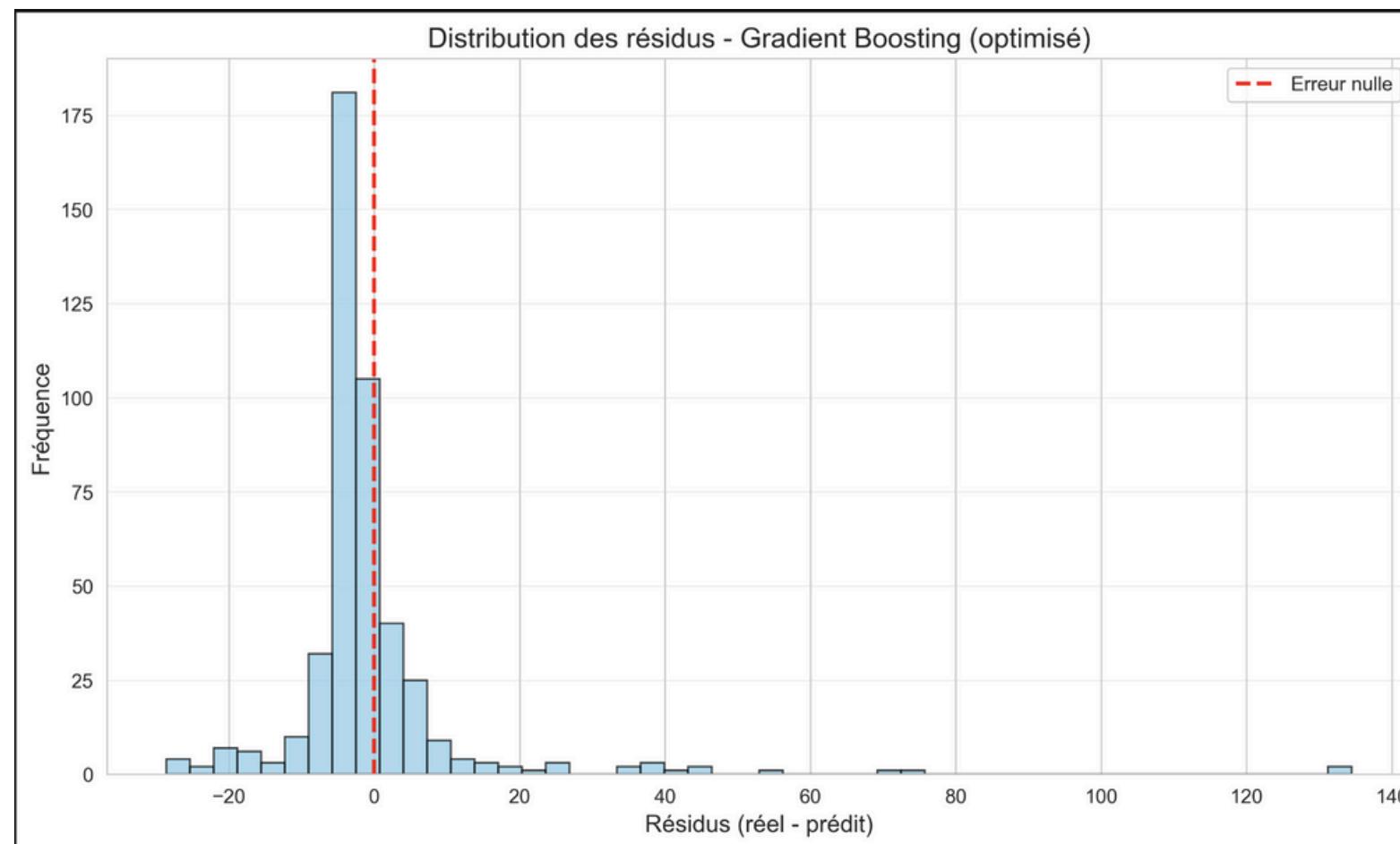
# Features importantes



### Top 5 features

1. `vote\_average` (27.2%)
2. `vote\_count` (14.7%)
3. `log\_vote\_count` (14.4%)
4. `years\_since\_release` (11.8%)
5. `original\_language\_encoded` (6.5%)

# Analyse des résidus



**Distribution centrée autour de 0**

Limitation : Quelques valeurs aberrantes (films très populaires mal prédis)

# *Synthèse des résultats*



**Dataset :** 2251 films nettoyés et préparés

**Analyse exploratoire :** Facteurs identifiés (année de sortie, vote\_average)

**Modélisation :** 6 modèles testés, Gradient Boosting optimisé = meilleur

**Performance :**  $R^2 = 0.154$ , RMSE = 13.64

**Limitations :** Difficulté à prédire les valeurs extrêmes

## 6. Résultats et conclusion

# *Conclusion et perspectives*

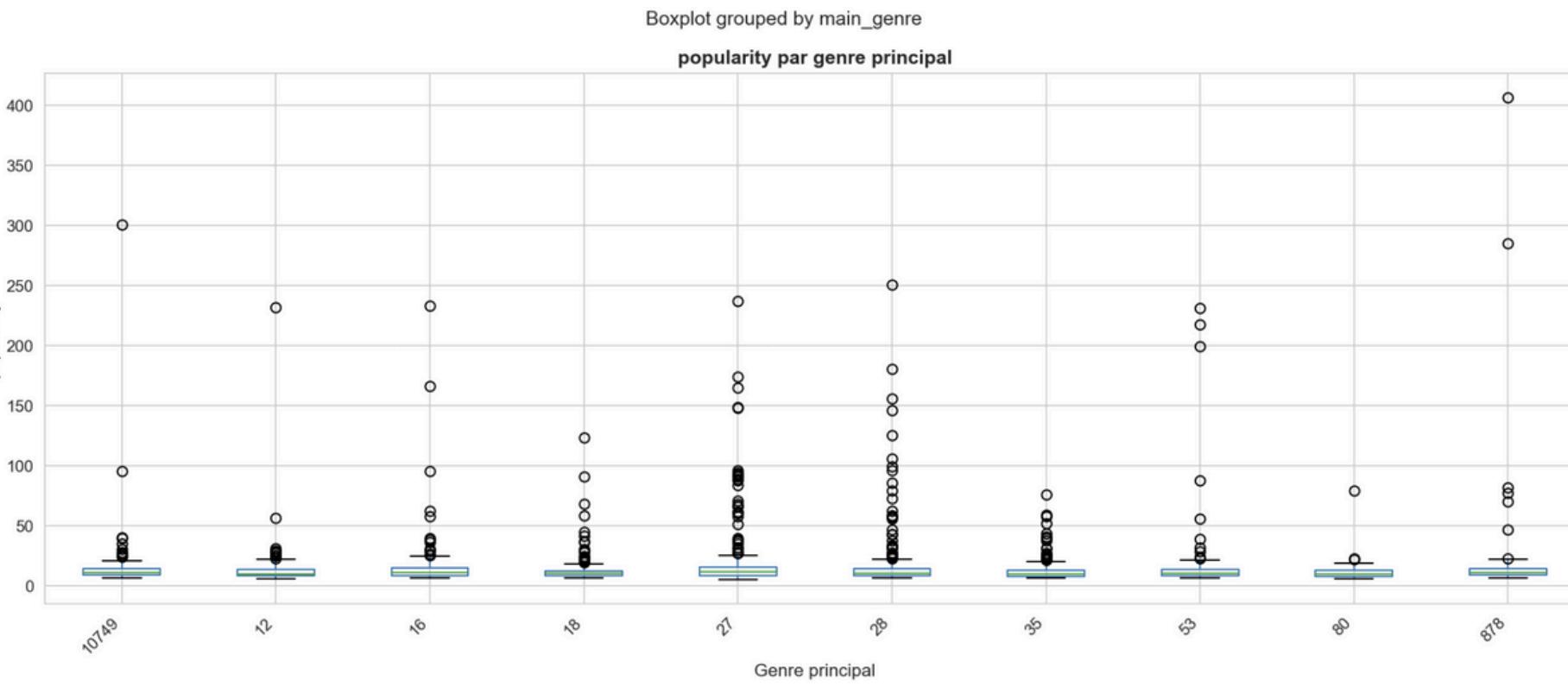
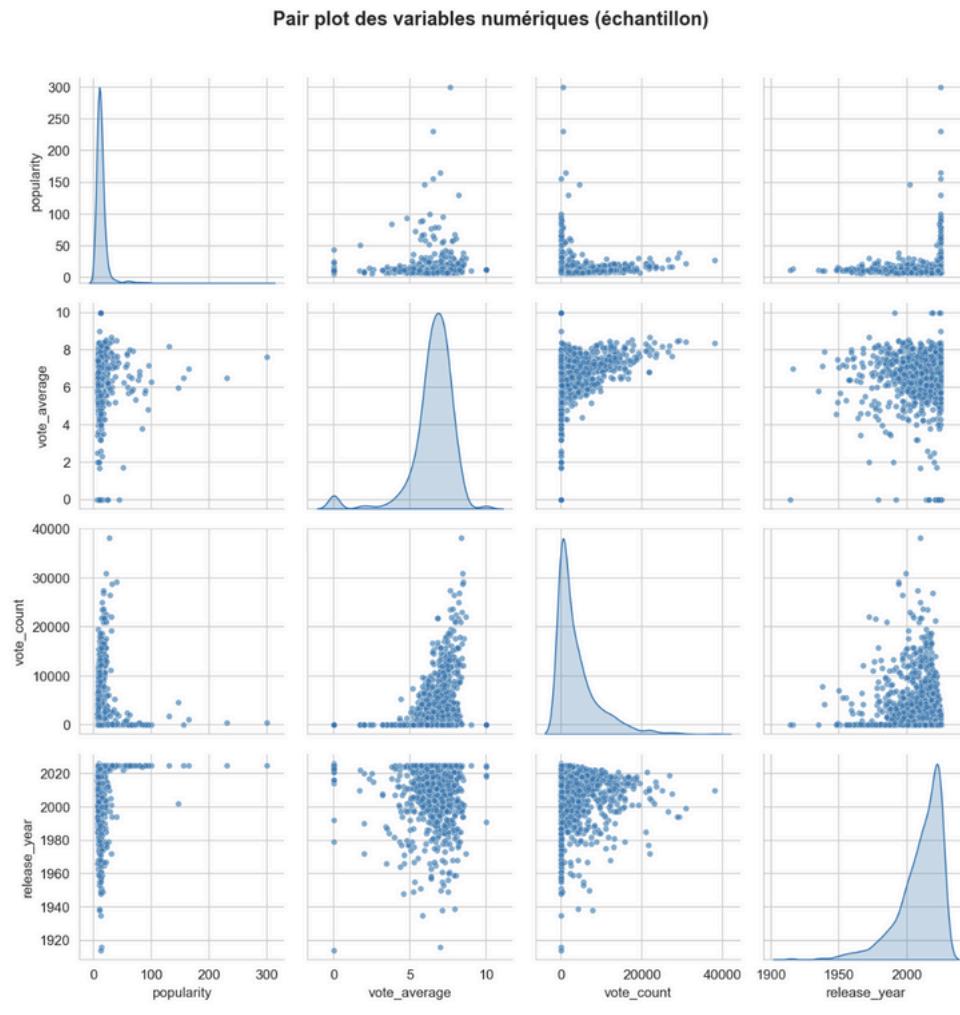
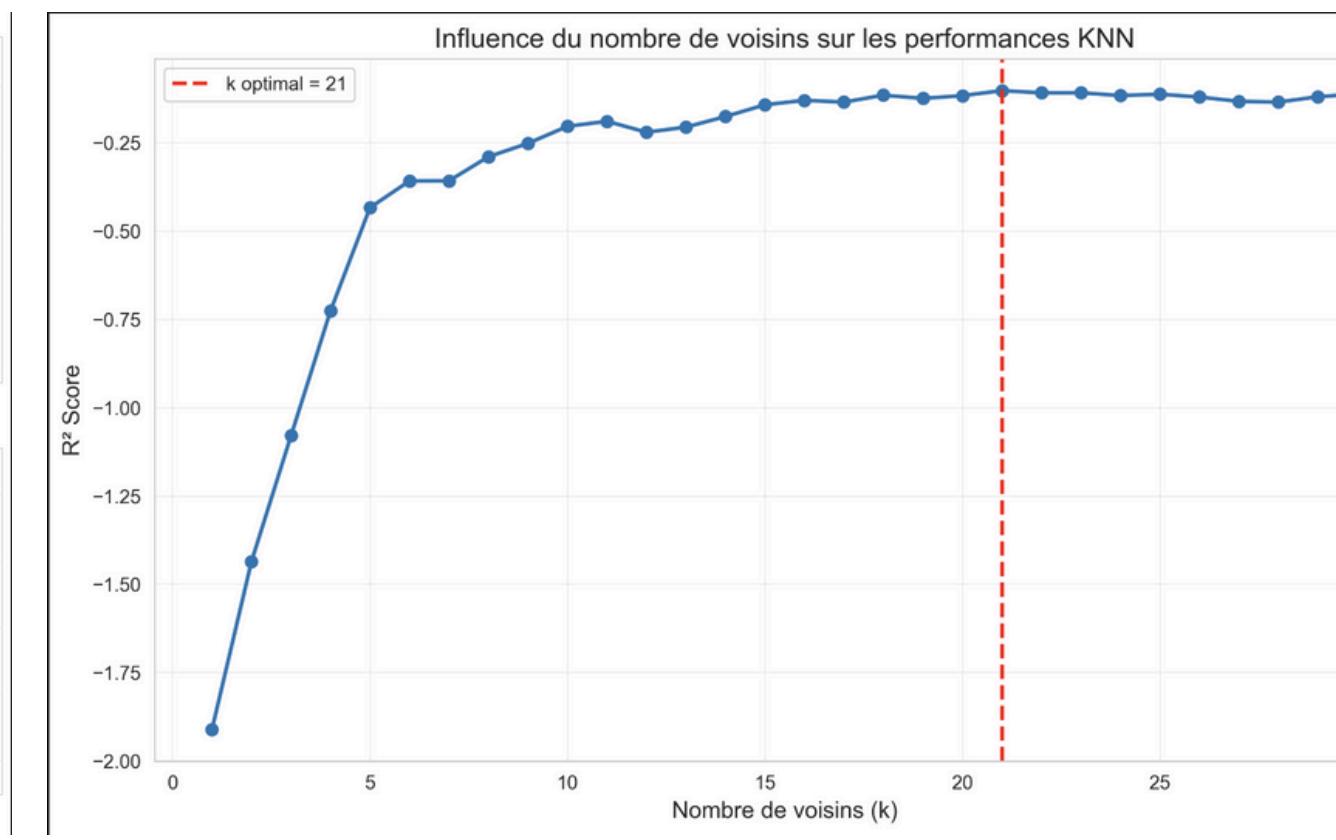
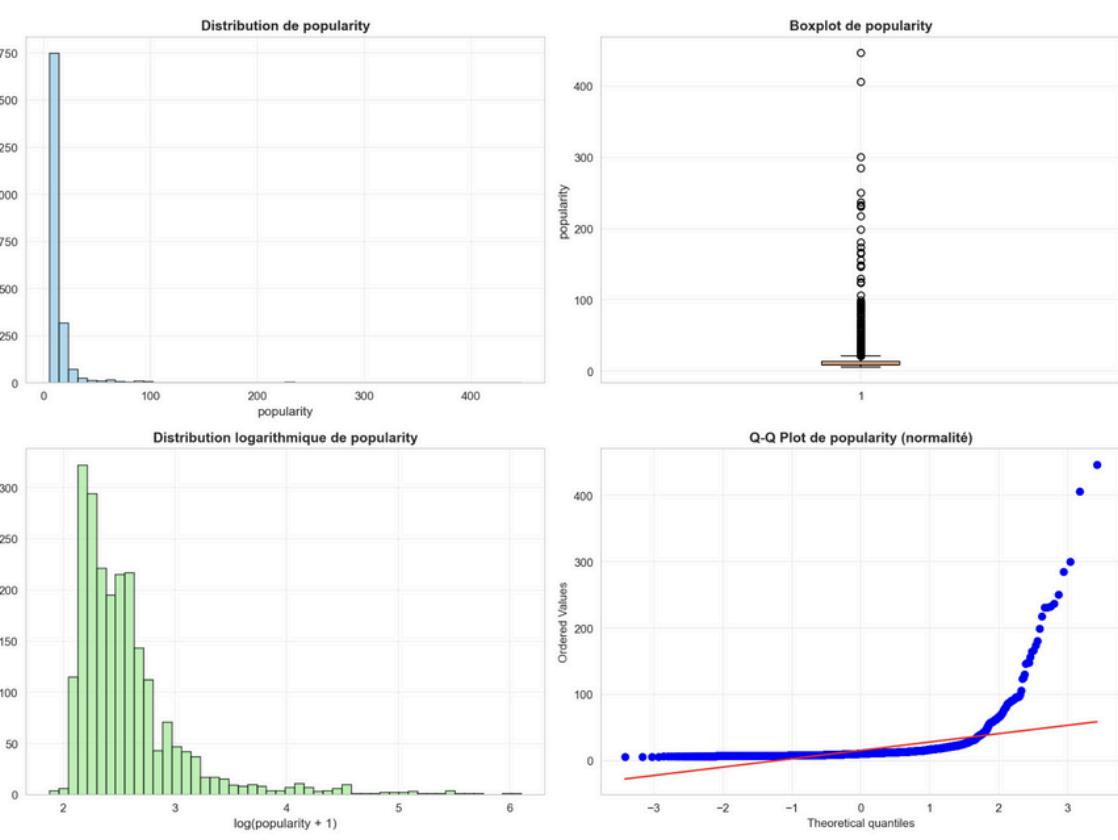
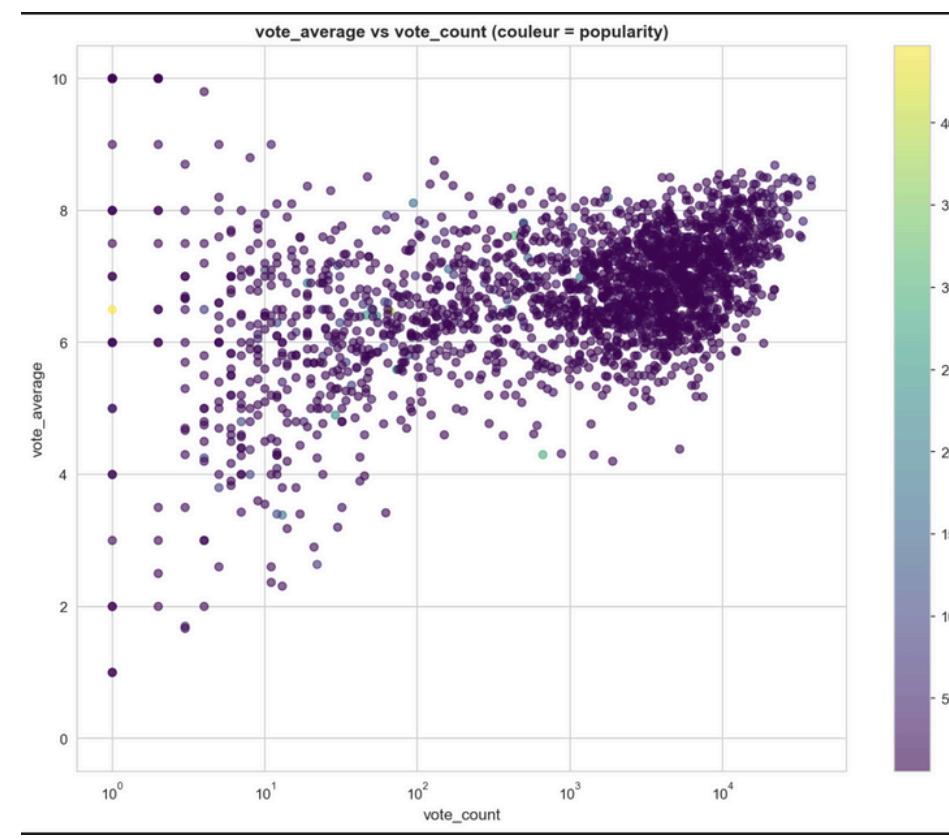


### Résultats clés :

- L'année de sortie et vote\_average sont les facteurs les plus importants
- Les modèles non-linéaires (Gradient Boosting) sont plus performants

### Perspectives :

- Ajouter des features (budget, acteurs, réalisateurs)
- Tester d'autres modèles (XGBoost, réseaux de neurones)
- Améliorer la prédiction des valeurs extrêmes



# *Questions ?*

