Prédire la popularité d'un animé

Anime Dataset

Arnaud GRASSIAN & Vithuson VAITHILINGAM

Groupe 2

Décembre 2025

Figure 1 – Couverture du Dataset Kaggle

Contexte

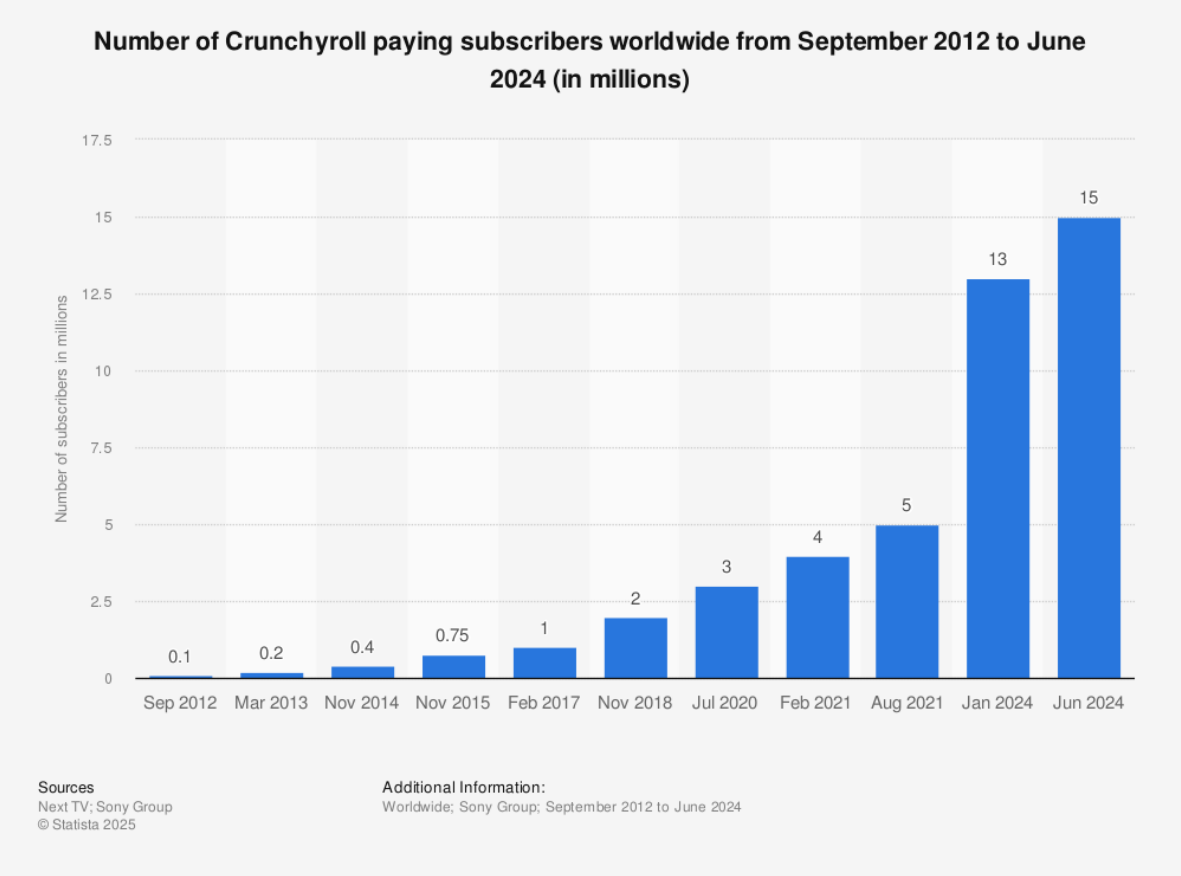

Figure 2 – Plateforme de streaming d'anime —Exemple de popularité.

Nature du problème

Problématique 1 : Prédire la note d'un anime

- Comprendre quels facteurs influencent la qualité perçue.
- Construire un modèle capable d'estimer le score moyen.

Problématique 2 : Recommander un anime à un utilisateur

- Exploiter les préférences et l'historique d'un utilisateur.
- Proposer des recommandations personnalisées.

Les 13 jeux de données du projet

Nom du fichier	Nb de lignes	Description
ratings.csv	124,298,357	Notes attribuées par les utilisateurs
favs.csv	4,178,747	Favoris déclarés par les utilisateurs
person_voice_works.csv	489,516	Rôles de doublage des personnes
person_anime_works.csv	458,091	Contributions des personnes aux animes
profiles.csv	337,155	Profils utilisateurs du dataset
character_anime_works.csv	236,816	Associations personnages –animes
characters.csv	208,727	Données sur les personnages
recommendations.csv	105,249	Recommandations d' animes
person_details.csv	76,699	Détails (staff/doubleurs)
character_nicknames.csv	36,923	Surnoms
details.csv	28,955	Métadonnées principales des animes
stats.csv	28,955	Statistiques (score, popularité)
person_alternate_names.csv	20,465	Noms alternatifs des personnes

Tableau récapitulatif des sources de données utilisées dans le projet.

One-Hot Encoding

- Transformer une variable catégorielle en variables numériques

Variable : `type` (*TV*, *Movie*, *OVA*, *ONA*)

Avant

type
TV

⟶

Après

TV	Movie	OVA	ONA
1	0	0	0

Exemple de One-Hot Encoding appliqué à la variable `type` du projet

Préparation et variables cibles

- Réduction des tables aux colonnes pertinentes.
- Échantillonnage pour rendre la modélisation compatible avec les limites mémoire.

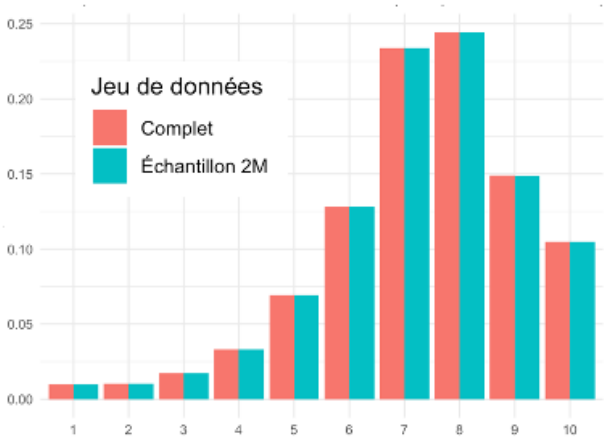

Figure 3 – Distribution des notes dans le tableau initial et échantillonné

Définition des deux variables cibles :

- **Score global** : note moyenne d'un anime.
- **Y_user_score** : Note qu'un utilisateur a attribué à un animé

Corrélation

Var1	Var2	Corr
status_Currently Airing	status_Finished Airing	-1.0000000
num_music_people	num_music_entries	1.0000000
type_TV	season_	-0.9992695
rating_Rx - Hentai	genre_Hentai	0.9964617
num_characters	num_supporting_characters	0.9933705
num_doubleurs	num_support_doubleurs	0.9905294
stream_CatchPlay	stream_MeWatch	0.9432780
sum_character_favorites	max_character_favorites	0.9389519
num_supporting_characters	num_support_doubleurs	0.7807730
num_characters	num_support_doubleurs	0.7721742

Table 1 – Paires de variables fortement corrélées dans le dataset

Modèles testés pour la note d'un anime

Forward, Backward & Stepwise : Problème mémoire

- Modèles pénalisés :

  On souhaite ici minimiser la fonction Φ donnée par :

  $$\Phi(\beta) = \|Y - X\beta\|_2^2 + (1-\alpha)\lambda\|\beta\|_2^2 + \alpha\lambda\|\beta\|_1^2 \qquad (\text{avec } \lambda \in \mathbb{R}_+^*)$$

  - **Ridge** : pénalisation ℓ_2, stabilise les coefficients et $\alpha = 0$
  - **Lasso** : pénalisation ℓ_1, sélection de variables et $\alpha = 1$
  - **Elastic Net** : combinaison $\ell_1 + \ell_2$, adapté aux variables corrélées et $\alpha \in]0, 1[$

- Validation croisée pour choisir le paramètre de régularisation λ (et α pour Elastic Net).

  On choisit λ parmi λ_{min} et λ_{1se}

Explication visuelle Elastic Net

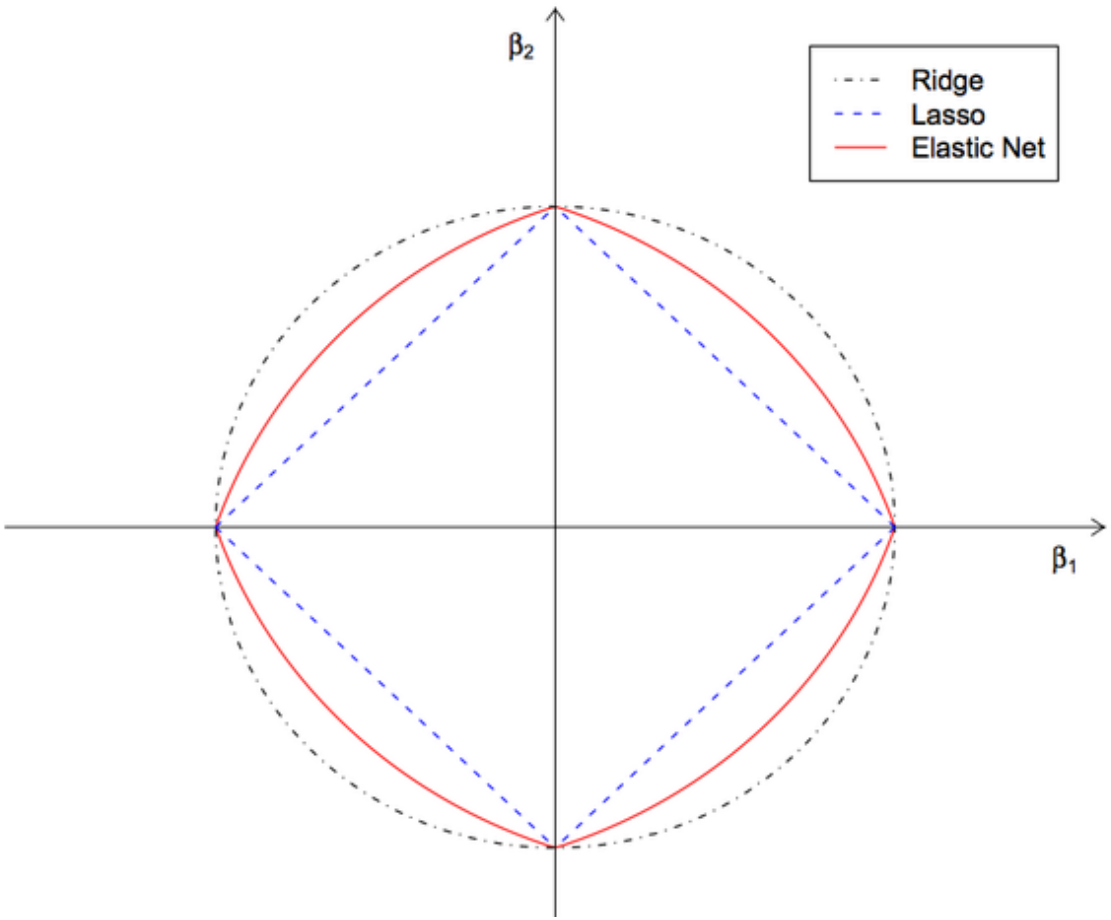

Figure 4 – Représentation géométrique des contraintes Ridge (ℓ_2), Lasso (ℓ_1) et Elastic Net

Validation croisée des modèles pénalisés

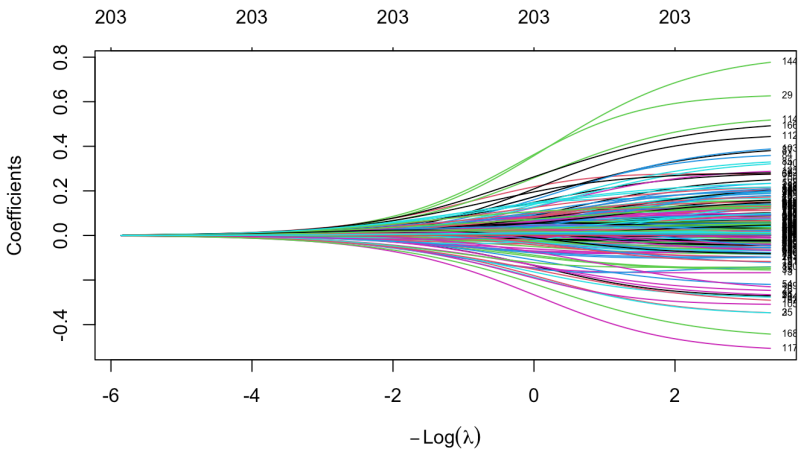

Figure 5 – Chemin Ridge

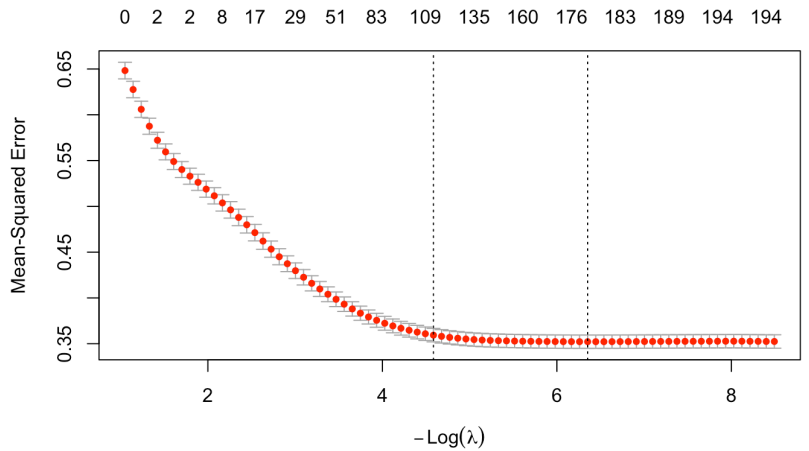

Figure 6 – Lasso Cross-Validation

- Pas de Surapprentissage (Annexes)

Comparaison des modèles et choix final

Modèle	**RMSE**	$\mathbf{R^2}$	$\boldsymbol{\lambda}$
OLS (standard)	0.6210	0.4621	-
Ridge (min)	0.6205	0.4629	0.0671192
Ridge (1se)	0.6265	0.4525	0.2973798
Lasso (min)	0.6205	0.4629	0.00174176
Lasso (1se)	0.6291	0.4480	0.01020152
Elastic Net (min)	0.6206	0.4628	0.00477895
Elastic Net (1se)	0.6323	0.4423	0.02550381

Table 2 – Comparaison des modèles avec leurs RMSE, R^2, et valeurs de λ.

Performances très proches pour Ridge, Lasso et Elastic Net (RMSE $\approx 0{,}62$, $R^2 \approx 0{,}46$). Elastic Net retenu :

- compromis entre stabilité (Ridge) et sélection de variables (Lasso),
- mieux adapté à la forte colinéarité induite par le One-Hot Encoding.

Qualité du modèle retenu

ID	Note_Reelle	Note_Predite	Ecart
1	5.43	6.29	-0.86
2	6.26	6.94	-0.68
3	5.43	6.44	-1.01
4	5.74	6.43	-0.69
5	6.52	7.23	-0.71

Table 3 – Comparaison entre notes réelles, notes prédites et écarts associés.

Statistique	Valeur
Min.	-4.18
1st Qu.	-0.38
Median	0.01
Mean	-0.00938
3rd Qu.	0.42
Max.	1.77

Table 4 – Statistiques descriptives des erreurs de prédiction.

Données pour la recommandation

- Objectif : prédire si un utilisateur va **aimer** un animé.
- Construction de la variable cible "liked" :
  - Classe **1** : l'utilisateur **aime** l'animé $\Rightarrow$ note réelle > 7
  - Classe **0** : l'utilisateur **n'aime pas / peu** l'animé $\Rightarrow$ note réelle ≤ 7

Modèle de régression logistique

$$\text{logit}\, p_\beta(x_i) = \ln\left(\frac{p_\beta(x_i)}{1 - p_\beta(x_i)}\right) = \beta_1 x_{i,1} + \cdots + \beta_p x_{i,p} = \beta^T x_i$$

Où la fonction de transfert est donnée par la **fonction sigmoïde** :

$$p_\beta(x_i) = \frac{e^{\beta^T x_i}}{1 + e^{\beta^T x_i}}$$

Résultats obtenus avec le modèle

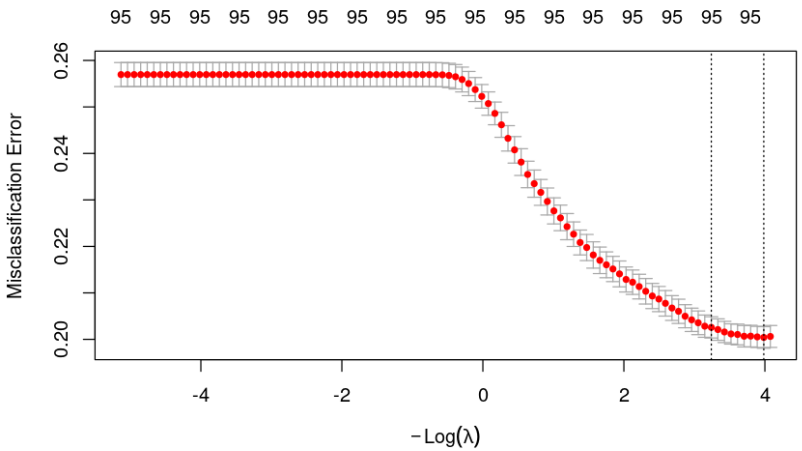

Figure 7 – Ridge

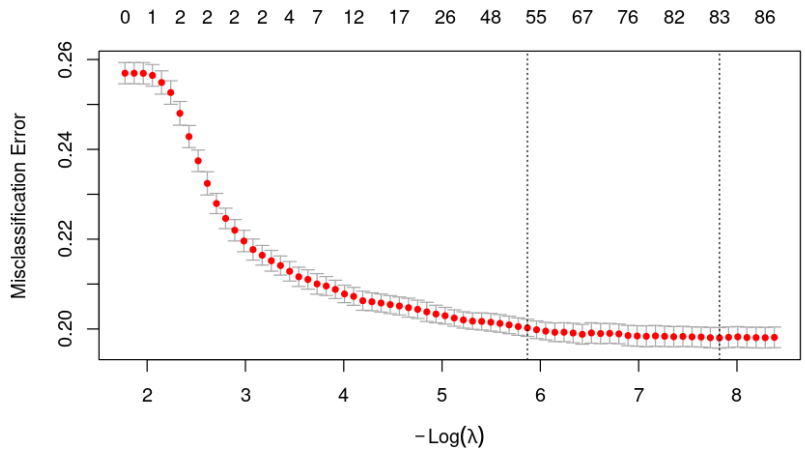

Figure 8 – Lasso

- Pas de Surapprentissage (Annexes)

Résultats des modèles

	Lasso (min)	Lasso (1se)	Ridge (min)	Ridge (1se)
Matrice de Confusion	$\begin{pmatrix} 1024 & 1685 \\ 425 & 7469 \end{pmatrix}$	$\begin{pmatrix} 1044 & 1705 \\ 442 & 7469 \end{pmatrix}$	$\begin{pmatrix} 949 & 1780 \\ 358 & 7536 \end{pmatrix}$	$\begin{pmatrix} 886 & 1843 \\ 299 & 7595 \end{pmatrix}$
Accuracy	0.7996	0.7993	0.7986	0.7982

	Elastic Net (min)	Elastic Net (1se)
Matrice de Confusion	$\begin{pmatrix} 1059 & 1670 \\ 456 & 7438 \end{pmatrix}$	$\begin{pmatrix} 970 & 1759 \\ 370 & 7424 \end{pmatrix}$
Accuracy	0.7999	0.7996

Table 5 – Comparaison des différents modèles

Poids des β

Variable	Poids
status_dropped	-2.9171844
user_mean_score_from_ratings	1.1628601
genre_Award.Winning	0.8477050
source_Mixed_media	-0.7696437
genre_Horror	-0.6341143
source_Game	-0.5670382
genre_Erotica	-0.5625920
status_Currently_Airing	0.5552745
genre_Girls.Love	-0.5244876
genre_Boys.Love	-0.4807013

Table 6 – Poids des variables dans le modèle.

Exemple de recommandation pour Death Note

Utilisateur	Score_Compatibilite
ReMightyRon	99.96
8angel	99.91
PerfectGod	99.89
THE_HIDDEN	99.89
SasakiMichie	99.83
BENBOURY	99.82
3MeowNeko3	99.79
60189134403	99.76
HerrscherOfFlame	99.73
Zer0Two002_	99.72

Table 7 – Top recommandations d'utilisateurs compatibles pour *Death Note*.

Conclusion

Résultats

- Prédiction de la note (sur 10) d'un anime avec une erreur moyenne de 0.6
- Recommandation des œuvres à des utilisateurs avec + de 80% de pertinence

Limites

- Fichiers trop volumineux, échantillonnage
- Beaucoup de données manquantes
- R^2=0.46 : nos variables n'expliquent que la moitié de la note
- Vérification de la conformité des données

Ouverture

- Tester des modèles basés sur des arbres de décision
- Traitement NLP pour analyser les synopsis

Annexes

Modèle	R2_Train	R2_Test	Écart
OLS (standard)	0.4832	0.4621	0.0211
Ridge (min)	0.4798	0.4629	0.0169
Ridge (l1se)	0.4673	0.4525	0.0148
Lasso (min)	0.4802	0.4629	0.0173
Lasso (l1se)	0.4612	0.4480	0.0132
Elastic Net (min)	0.4802	0.4629	0.0173
Elastic Net (l1se)	0.4547	0.4423	0.0124

Table 8 – Performances des modèles pénalisés pour anime_level

Modèle	Acc_Train	Acc_Test	Écart
Ridge (l1se)	0.7985	0.7985	0.0000
Lasso (l1se)	0.8000	0.7997	0.0003
Elastic Net (l1se)	0.8007	0.8000	0.0007
Ridge (min)	0.8009	0.7996	0.0013
Elastic Net (min)	0.8017	0.8001	0.0017
Lasso (min)	0.8018	0.8000	0.0018

Table 9 – Performances des modèles pénalisés pour user_anime_level