

Introduction :

L'objectif de ce projet est **double** : modéliser le succès critique d'un anime et prédire l'affinité d'un utilisateur pour un contenu spécifique.

Les jeux de données (anime_level.csv et user_anime_level.csv) comportant un grand nombre de valeurs manquantes, nous avons appliqué un protocole strict : suppression des colonnes puis des lignes contenant plus de 50% de NA. Nous avons utilisé un échantillon d'entraînement (~80%) et d'un échantillon de test (~20%).

Pour user_anime_level.csv, nous modélisons la probabilité d'appréciation d'une œuvre par un utilisateur en construisant une variable de réponse binaire « Liked », prenant la valeur 1 si la note attribuée atteint le seuil de satisfaction de 7/10, et 0 sinon.

Absence de modèle linéaire simple

Contrairement à la démarche classique consistant à débuter par un modèle linéaire, cette approche n'a pas pu être appliquée ici en raison de la taille très importante du jeu de données, amplifiée par le one-hot encoding des variables catégorielles. Pour ces raisons, la modélisation a directement commencé par des méthodes régularisées, mieux adaptées à ce type de données à grande dimension.

Prédiction du Score Global

Nous cherchons à expliquer la variable continue Y_score_global (note moyenne sur MyAnimeList) par les caractéristiques intrinsèques de l'œuvre (Genre, Studio, Source, etc.).

Nous avons comparé trois approches de régression pénalisée :

- Ridge ($\alpha = 0$) : Conserve toutes les variables en réduisant les coefficients,
- Lasso ($\alpha = 1$) : Effectue une sélection de variables (coefficients mis à zéro).

Pour ces 2 modèles, une validation croisée a été utilisée afin d'optimiser λ .

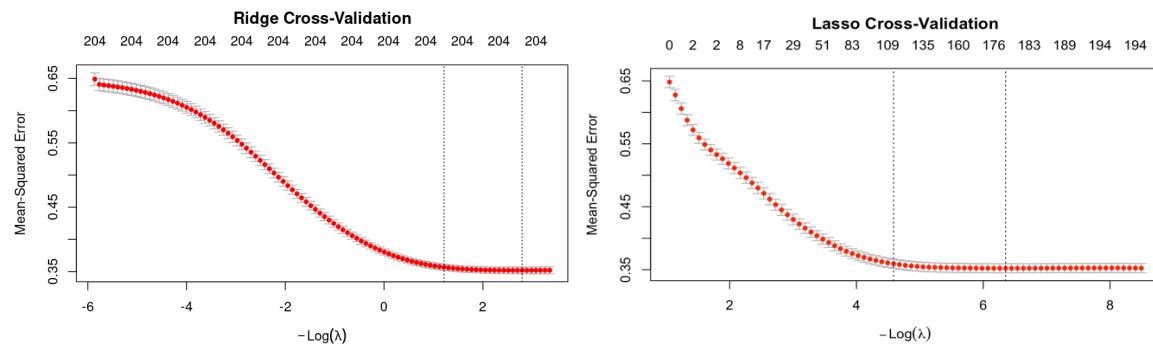


Figure 1 - Erreur de validation croisée de Ridge et Lasso en fonction de λ , avec indication de λ_{\min} et λ_{1se} .

Pour le Lasso, la validation croisée montre que l'erreur diminue rapidement lorsque la pénalisation se relâche, avant de se stabiliser autour d'une valeur optimale. Pour Ridge, la décroissance de l'erreur est plus progressive et plus régulière. Dans les deux cas, deux valeurs de λ ont été conservées : λ_{\min} (minimise l'erreur de validation croisée) et λ_{1se} (solution plus régulière, dans un écart-type du minimum). Ces valeurs délimitent la zone où les modèles atteignent leurs meilleures performances

- Elastic Net : ce modèle combine les avantages de Ridge et de Lasso, en contrôlant la proportion de pénalisation ℓ_1 / ℓ_2 via un paramètre α . Une recherche sur grille ($\alpha \in [0.1, 0.9]$) a été menée pour optimiser le compromis biais-variance.

Modèle	RMSE	R^2
Ridge (min)	0.6205	0.4629
Ridge (1se)	0.6265	0.4525
Lasso (min)	0.6205	0.4629
Lasso (1se)	0.6291	0.4480
Elastic Net	0.6206	0.4628

Tableau 1 - RMSE et R^2 des modèles

Les différents modèles testés (Ridge, Lasso et Elastic Net) présentent des performances très proches, avec des valeurs de RMSE comprises entre 0.6205 et 0.6291 et un coefficient de détermination $R^2 \approx 0.46$. Les écarts observés sont donc faibles et ne permettent pas de distinguer nettement un modèle. Dans ce contexte, le choix s'est porté sur **Elastic Net**, qui combine les avantages de Ridge et Lasso.

Une analyse des résidus montre une dispersion homogène autour de la diagonale (prédiction vs réalité), ainsi qu'une distribution des erreurs centrée, sans biais marqué.

Figure 2 :

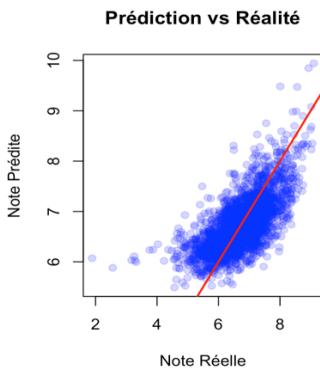
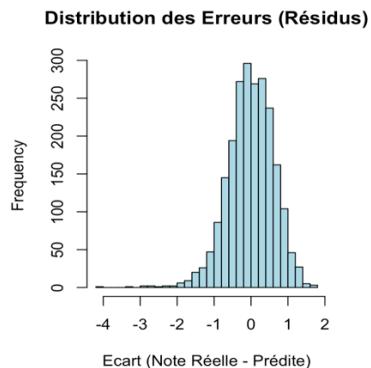


Figure 3 :



Recommandation d'un anime

Contrairement à la prédiction du score global, notre cible est ici binaire. Nous avons donc opté pour une Régression Logistique (famille binomial dans `glmnet`), qui constraint la prédiction à une probabilité entre 0 et 1. L'ajout de la pénalisation (ℓ_1 / ℓ_2) est indispensable ici en raison de la dimension élevée du tableau : le one-hot encoding des variables utilisateurs et animes génère un grand nombre de colonnes, risquant un sur-apprentissage massif sans régularisation.

Modèle	Accuracy
Lasso (min)	0.7998
Elastic Net	0.7998
Lasso (1se)	0.7995
Ridge (min)	0.7987
Ridge (1se)	0.7984

Tableau 2 - Accuracy des modèles

Les performances des différents modèles sont quasiment identiques, avec une accuracy $\approx 79.9\%$. Nous retenons le modèle Lasso (1se) : même s'il n'est pas strictement le meilleur en termes d'accuracy, l'écart avec le premier modèle est insignifiant (< 0.0003). En revanche, sa pénalisation ℓ_1 plus forte permet de supprimer davantage de variables non pertinentes, ce qui en fait l'option la plus robuste et la plus interprétable dans un contexte où le nombre de variables est très élevé (plus que dans l'autre dataset).

La matrice de confusion montre que le modèle distingue très bien la classe 1 (rappel ≈ 0.946), ce qui signifie qu'il identifie correctement la quasi-totalité des utilisateurs qui aiment un anime. En revanche, le rappel de la classe 0 est beaucoup plus faible (≈ 0.375), indiquant une difficulté à détecter les utilisateurs qui n'aiment pas un anime.

Prédiction		
Réalité	0	1
0	1024	1705
1	425	7469
Métrique		
Accuracy	0.799	
Recall classe 0	0.375	
Recall classe 1	0.946	

Tableau 3 - Métriques du Lasso 1se

Dans notre contexte, cette asymétrie est en réalité souhaitable : l'objectif principal est de ne pas manquer les animes qu'un utilisateur est susceptible d'aimer. Il vaut mieux recommander un peu trop d'animes pertinents que de passer à côté d'animes qu'un utilisateur apprécierait fortement.

Conclusion :

Les méthodes de régression pénalisée fournissent donc un cadre robuste pour prédire à la fois la qualité d'un anime et les préférences des utilisateurs, tout en assurant une bonne interprétabilité des résultats. Elles constituent une base solide pour la mise en place d'un système de recommandation efficace et cohérent.