

# **OPMA 419: Group Project Report – Animes**

Daniel Caubalejo, Aaron Diwa, Carlton Mah

Haskayne School of Business, University of Calgary

OPMA 419: Predictive Models in Business Analytics

Professor Alireza Sabouri

April 9, 2024

## **Problem and Goal Statement**

Anime, a style of Japanese animated entertainment, has been steadily growing in the Western space for a few years. Currently, the anime market is valued at USD 26 billion and is expected to grow to USD 63 billion by 2032 with a compound annual growth rate of around 9%. (Singh, 2024). These statistics imply a growth in new anime watchers. Since anime, as a medium, has existed for more than four decades, with millions of anime released throughout the years, the problem arises for new anime watchers: How can they find shows that are proper introductions to the genre? Through this problem, our team set a goal: Make a model that makes it easier for anime watchers to find shows that they may like.

## **Data Mining Task**

Our main objective is to create a predictive model that forecasts the overall score of an anime based on genre, episode count, number of members, ranking, and popularity predictors. All our models will be using a seed of “2024” for consistent results. The score attribute ranks from 0 to 10, where a lower number can depict an anime to be less enjoyable, and a higher number can describe an anime to be more enjoyable. The closer an anime scores to 10, the more the user of the model can expect to enjoy watching the anime.

## **Data Discussion**

Our anime dataset was pulled from Kaggle, where the data originates from MyAnimeList (MAL). For context, MAL is a community run anime and manga database that provides a scoring and organization system for various anime. The dataset has been updated as of March 3<sup>rd</sup>, 2024 and contains 19,311 records in total with the columns: uid, title, synopsis, genre, aired, episodes, members, popularity, ranked, score, img\_url and link. Of these attributes, the independent

variables are genre, episodes, members, ranking and popularity. The dependent variable or the Y-variable is score, and will be used as our label in RapidMiner.

## Exploratory Analysis

Limiting our analysis to 1,000 records was important in analyzing the data due to the large number of records in the dataset. This was done through the sample and normalize operators in RapidMiner and was a crucial step in relieving hardware limitations and run time of our processes. Most importantly, sampling and normalizing allows us to maintain a similar distribution as the original dataset and ensure our analysis was performed on the same scale. As seen in the histograms below, our distributions before and after sampling appear to be similar.

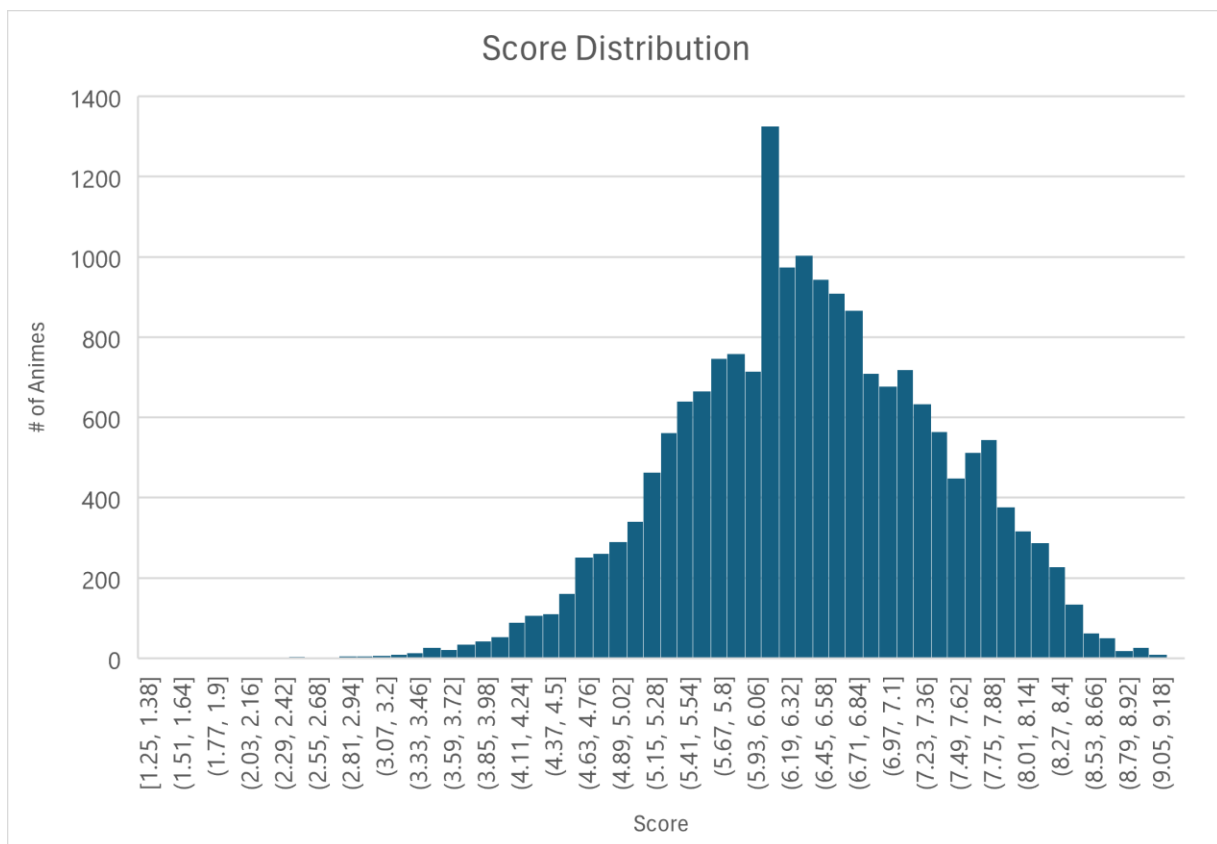


Figure 1: Histogram of score distribution before sampling.

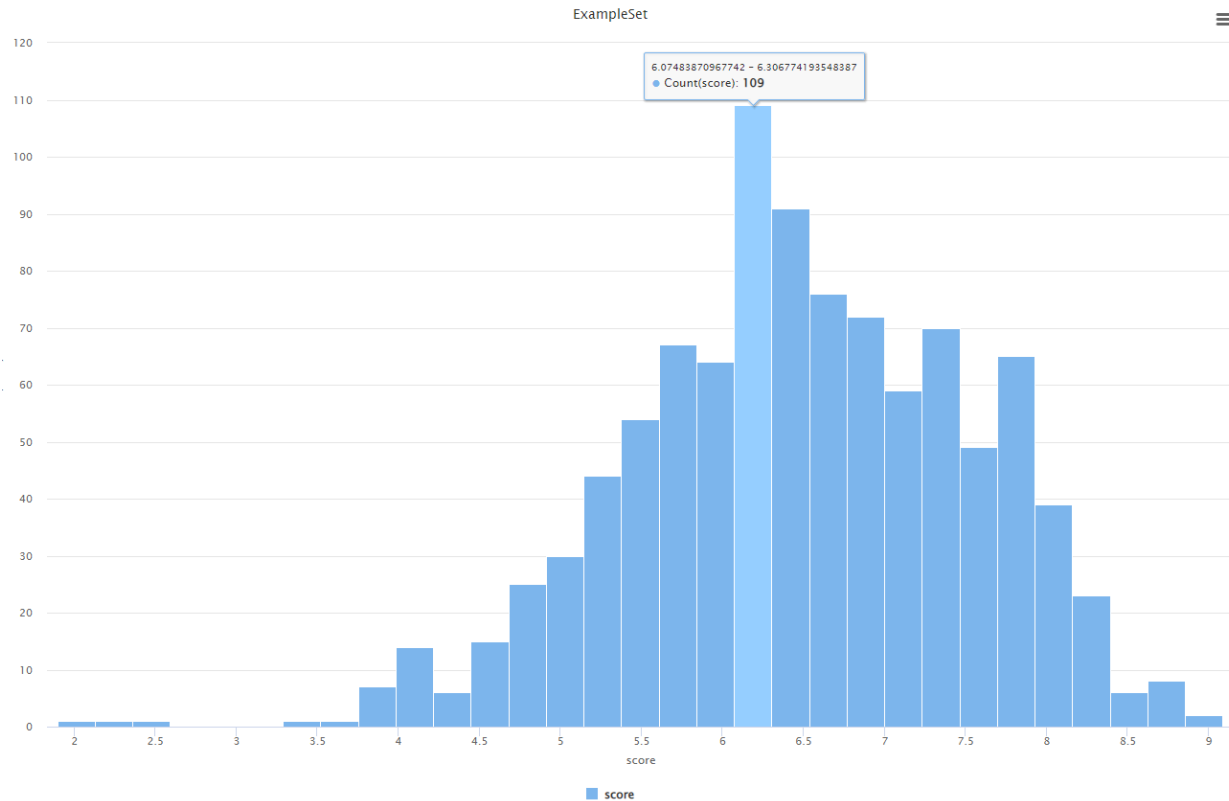


Figure 2: Histogram of score distribution after sampling 1,000 records.

A trend of ongoing anime missing information was also noted, as many popular and currently airing anime were missing predictor data such as episode number. This is important to consider for our analysis as some of the most popular and modern anime belonged to this list. Since these animes were missing attribute data, they had to be omitted from our analysis, potentially skewing the data.

The cleaning process involved removing records with missing attribute data, removing irrelevant columns and splitting our genre column into independently functional columns. The purpose of the cleaning process was to transform our anime spreadsheet into a dataset that is both usable and easily comprehensible for analysis. Cleaning also speeds up Rapid Miner's run time as more records slow the process down. Pertaining to removing irrelevant columns, we found synopsis, aired, img\_url and link to not hold useful information when creating our models. If we

had kept these columns in, we would only be introducing additional noise to our data and potentially broken or unreliable data. An example being the url of an image or link of an anime changing. Therefore, we decided it was best to remove these columns as they seemed irrelevant to our goal and to save processing power for more significant attributes.

Splitting our genre column was necessary as RapidMiner would interpret several genres found in a single Excel cell to be a single unique genre (see figure 3 and 4). We wanted to avoid this issue as our model should make predictions based on individual genres, rather than relying on the specific sequence of genres. Additionally, since some animes were classified into multiple genres, extending up to 13 in certain cases, so we decided to limit the genre columns to the top 3 genres per record. By doing this, we can reduce RapidMiner's process time and avoid hardware bottlenecks. Most importantly, splitting up the genre column as seen in figure 4, allows us to independently utilize each genre in our analysis. Finally, our cleaned dataset can be seen in figure 6 below.

genre
['Comedy', 'Sports', 'Drama', 'School', 'Shounen']
['Drama', 'Music', 'Romance', 'School', 'Shounen']
['Sci-Fi', 'Adventure', 'Mystery', 'Drama', 'Fantasy']

Figure 3: Genre column before cleaning, most relevant genres listed first in order of priority.

genre1 ▼	genre2 ▼	genre3 ▼
Comedy	Sports	Drama
Drama	Music	Romance
Sci-Fi	Adventure	Mystery

Figure 4: Genre columns after splitting, RapidMiner interprets them all independently.

uid	title	synopsis	genre	aired	episodes	members	popularity	ranked	score	img_url	link	
28891	Haikyuu!!	Followin	['Comedy', 'Sports	Oct 4, 201	25	489888	141	25	8.82	https://cd	https://myanimelist.net/	
23273	Shigatsu v	Music	['Drama', 'Music',	Oct 10, 20	22	995473	28	24	8.83	https://cd	https://myanimelist.net/	
34599	Made in A The		['Sci-Fi', 'Adventur	Jul 7, 2017	13	581663	98	23	8.83	https://cd	https://myanimelist.net/	
5114	Fullmetal	"In order	['Action', 'Military	Apr 5, 200	64	1615084	4	1	9.23	https://cd	https://myanimelist.net/	
31758	Kizumono	After	['Action', 'Mystery	06-Jan-17	1	214621	502	22	8.83	https://cd	https://myanimelist.net/	
37510	Mob Psycl	Shigeo	['Action', 'Slice of	Jan 7, 201	13	442310	176	21	8.89	https://cd	https://myanimelist.net/	
199	Sen to Chi	Stubborn	['Adventure', 'Sup	20-Jul-01	1	913212	40	20	8.9	https://cd	https://myanimelist.net/	
38000	Kimetsu n	Ever	['Action', 'Demon	Apr 6, 201	26	575037	106	19	8.92	https://cd	https://myanimelist.net/	
35247	Owarimor	Followin	['Mystery', 'Comed	Aug 12, 20	7	189944	573	18	8.93	https://cd	https://myanimelist.net/	

Figure 5: Dataset before cleaning.

uid	title	genre1	genre2	genre3	episodes	members	popula	ranked	score
28891	Haikyuu!!	Comedy	Sports	Drama	25	489888	141	25	8.82
23273	Shigatsu v	Drama	Music	Romance	22	995473	28	24	8.83
34599	Made in A	Sci-Fi	Adventure	Mystery	13	581663	98	23	8.83
5114	Fullmetal	Action	Military	Adventure	64	1615084	4	1	9.23
31758	Kizumono	Action	Mystery	Supernatu	1	214621	502	22	8.83
37510	Mob Psycl	Action	SliceofLife	Comedy	13	442310	176	21	8.89
199	Sen to Chi	Adventure	Supernatu	Drama	1	913212	40	20	8.9
38000	Kimetsu n	Action	Demons	Historical	26	575037	106	19	8.92
35247	Owarimor	Mystery	Comedy	Supernatu	7	189944	573	18	8.93

Figure 6: Dataset after cleaning.

## Data Partitioning and Dimension Reduction

In RapidMiner, all our models followed the same data preparation process. First, we start off with setting our label and id with the “Set Role” operator, making “score” our label and “title” our id. The label “score” will be our predicted variable. We then create our dummy variables and comparison groups using the highest counts of genres within the genre1, genre2 and genre3 columns (see figure 8). The “Select Attributes” variable was then used to exclude uid from our analysis as our id “title” would now indicate which anime belonged to each analyzed record. And finally, the “Sample” operator was used to limit the number of records processed as 1,000, while keeping the distribution of our original dataset the same.

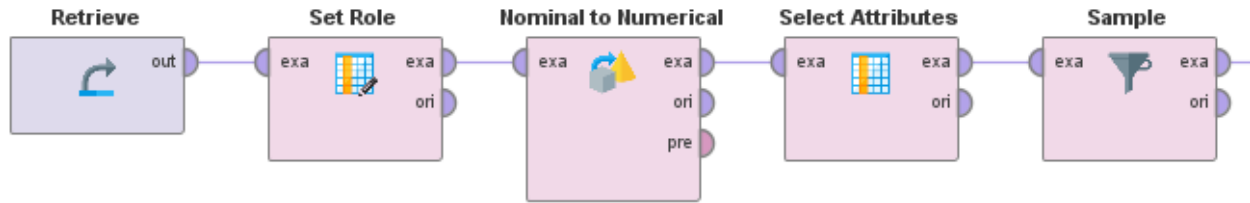


Figure 7: Data prep process for all models.

comparison group attribute	comparison group
genre1	▼ Action
genre2	▼ Comedy
genre3	▼ Fantasy

Figure 8: Nominal to Numerical operator comparison groups.

Regarding partitioning of data, we followed a 70/30 training and validation split, with 70% for training and 30% for validation. Our seed was set to 2024 to ensure consistent and replicable results. Our CART, Random Forest and k-NN models utilized backwards elimination for dimension reduction as they were the only models in our testing that reduced RMSE and increased the  $R^2$  in their results. The purpose of using backwards elimination was to optimize our model in a way that uses variables that minimized the RMSE our model produced while eliminating attributes that were less significant.

attribute	weight ↑
genre1 = Comedy	0
genre1 = Mecha	0
members	0
popularity	0

Figure 9: k-NN backwards elimination removed variables.

attribute	weight ↑
genre1 = Kids	0
genre2 = Sci-Fi	0
genre2 = School	0

Figure 10: CART backwards elimination removed variables.

attribute	wei... ↑
genre1 = Supernatural	0
genre2 = Josei	0

Figure 11: Random Forest backwards elimination removed variables.

## Analysis

Model	RMSE	R <sup>2</sup>
<b>Linear Regression</b>	<b>0.543</b>	<b>0.719</b>
<b>K-NN</b>	<b>0.737</b>	<b>0.462</b>
<b>CART</b>	<b>0.414</b>	<b>0.834</b>
<b>Random Forest</b>	<b>0.491</b>	<b>0.762</b>
<b>Neural Networks</b>	<b>0.432</b>	<b>0.814</b>

## Model Testing

We used a variety of different models to evaluate our data evaluating primarily for root mean squared error (RMSE) and squared correlation (R<sup>2</sup>). RMSE measures the average difference between the predicted score dependent variable and the actual score. A lower RMSE



meant a closer fit with the actual score. Similarly, R2 measures the proportion of variation in the dependent variable (score) that can be attributed to the independent variables. R2 values lie between a range of 0-1 and a number closer to 1 indicates a closer fit between the data and the model (goodness of fit). Moreover, the models we analyzed were as follows: linear regression, k-NN, CART, random forest, and neural Networks. We will take a closer look at each of these in the following sections.

## Linear Regression

We first decided to implement Linear Regression as one of our models as it is relatively easy to set up, and since our y is a numerical value, linear regression seems to make sense.

Attribute	Coefficient	Std. Error	Std. Coefficient	Tolerance	t-Stat	p-Value	Code
genre3 = SliceofLife	0.012	0.019	0.012	0.999	0.640	0.523	
genre3 = Seinen	-0.028	0.022	-0.023	1.000	-1.239	0.216	
genre3 = Ecchi	0.004	0.017	0.004	1.000	0.230	0.819	
genre3 = Thriller	0.000	?	?	1	?	?	
genre3 = MartialArts	0.015	0.029	0.010	0.999	0.524	0.601	
genre3 = Psychol...	0.024	0.018	0.025	0.993	1.328	0.185	
genre3 = Shounen...	0.007	0.021	0.006	0.999	0.317	0.751	
genre3 = Dementia	0.004	0.024	0.003	0.999	0.169	0.866	
genre3 = Parody	0.005	0.019	0.005	1.000	0.252	0.801	
genre3 = Kids	0.099	0.020	0.092	0.992	4.978	0.000	****
genre3 = Harem	-0.007	0.022	-0.005	1.000	-0.297	0.767	
genre3 = Cars	-0.003	0.029	-0.002	0.999	-0.109	0.914	
genre3 = Vampire	-0.017	0.017	-0.019	0.998	-1.017	0.309	
genre3 = Samurai	0.038	0.021	0.034	0.998	1.828	0.068	*
genre3 = Game	0.009	0.024	0.007	0.999	0.371	0.711	
episodes	-0.025	0.018	-0.027	0.993	-1.430	0.153	
members	0.124	0.021	0.119	0.831	5.860	0.000	****
popularity	0.107	0.036	0.098	0.308	2.943	0.003	***
ranked	-0.933	0.037	-0.857	0.292	-25.056	0	****
(Intercept)	6.445	∞	?	?	0	1	

Figure 12: Attribute significance – Coefficients and p-values.

Note: the high coefficient of 'ranked' is probably due to the correlation between ranked and score (as score goes up ranked goes lower until it reaches 1, which in context, rank 1 should be the 'highest' rank, but lowest number.)

We then utilized optimized parameters on the Linear Regression operator, optimizing feature selection:

iteration	Linear Regression (2).feature...	root_m...
5	Iterative T-Test	0.544
4	T-Test	0.543
2	M5 prime	0.564
1	none	0.565
3	greedy	0.555

Figure 14: Linear Regression Optimized Parameters

As seen from the optimized parameters list, our results are as follows:

**root\_mean\_squared\_error**

`root_mean_squared_error: 0.543 +/- 0.000`

**squared\_correlation**

`squared_correlation: 0.719`

Figure 15: Linear Regression RMSE and R2.

## K-NN

We chose to use K-NN due to its flexibility and the fact that we have a large dataset.

After using Optimized Parameters on k, the results are the following with a k of 6:

### PerformanceVector

```
PerformanceVector:
root_mean_squared_error: 0.737 +/- 0.000
squared_correlation: 0.462
```

Figure 16: K-NN RMSE and R2.

## CART

The CART model was considered due to its ease of interpretability and automatic variable selection and reduction. It also allowed us to easily find that “ranked” was our most significant variable in the model. The CART model returned a RMSE of 0.414 with a R2 of 0.834.

### PerformanceVector

```
PerformanceVector:
root_mean_squared_error: 0.414 +/- 0.000
squared_correlation: 0.834
```

Figure 17: CART RMSE and R2.

The minimal leaf size parameter within the “Optimize Parameters” operator include an iteration range of 1-100 in steps of 100. This range combined with number of steps allowed us to get a minimized RMSE. Any iterations below or higher than 100 with a change in steps would result in an increased RMSE in comparison to our final result.

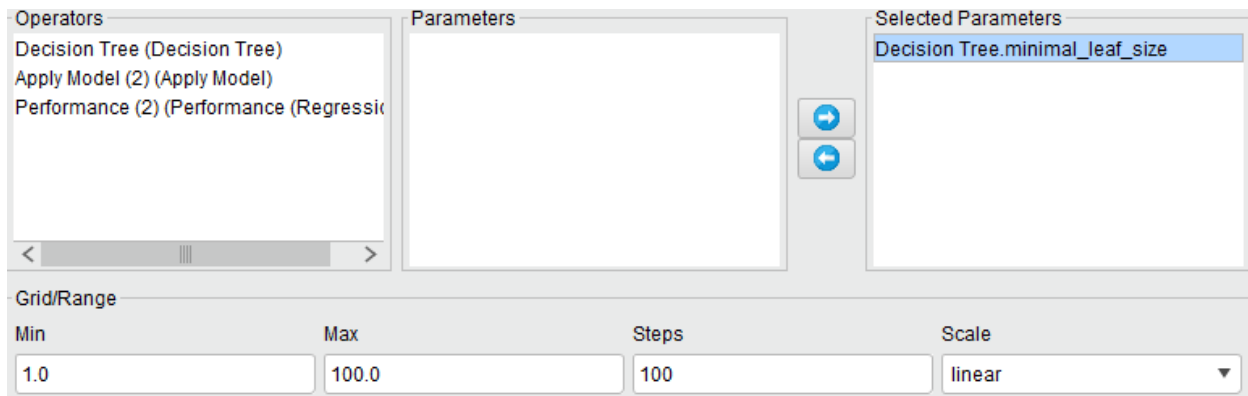


Figure 18: CART Optimize Parameters.

Our decision tree parameters include a least square criterion with a maximal depth of -1. We chose a maximal depth of -1 to allow the decision tree to infinitely grow until it fits the model. By utilizing infinite growth, we can minimize our RMSE.

As seen in the radial view of our CART model decision tree, we can easily find that “ranked” holds most significance in making a prediction. It is the predictor in the middle of the radial tree visualization.



Figure 19: CART Radial view decision tree.

Decision Tree (2) (Decision Tree)	
criterion ✓	least_square ▼ ⓘ
maximal depth ✓	-1 ⓘ
<input checked="" type="checkbox"/> apply prepruning ⓘ	
minimal gain ✓	0.01 ⓘ
minimal leaf size	2 ⓘ
minimal size for split	4 ⓘ
number of prepruning alternatives	3 ⓘ

Figure 20: CART Decision tree parameters.

Out of all the models we have tested, the CART model performed the best in predicting the scores of each anime. It had the lowest RMSE with a high R2 value compared to other models, therefore making it our recommended model for this dataset.

## Random Forest

The Random Forest model was considered due to its resistance to overfitting and stability with predictions. As seen below this model returns a RMSE of 0.529 and a R2 of 0.720.

**root\_mean\_squared\_error**

root\_mean\_squared\_error: 0.491 +/- 0.000

**squared\_correlation**

squared\_correlation: 0.762


Figure 21: Random Forest RMSE and R2.

The number of trees parameter within the “Optimize Parameters” operator include an iteration range of 1-29 in steps of 10. This range combined with number of steps allowed us to get a minimized RMSE. Any iterations below or higher than 29 would result in an increased RMSE compared to our final result.

The screenshot shows a software interface for optimizing Random Forest parameters. It consists of three main panels at the top: 'Operators', 'Parameters', and 'Selected Parameters'. The 'Operators' panel lists 'Random Forest (Random Forest)', 'Apply Model (2) (Apply Model)', and 'Performance (2) (Performance (Regression))'. The 'Parameters' panel is currently empty. The 'Selected Parameters' panel shows 'Random Forest.number\_of\_trees' selected. Below these panels is a 'Grid/Range' section with four input fields: 'Min' (1.0), 'Max' (29), 'Steps' (10), and 'Scale' (linear). There are also two blue arrows between the 'Parameters' and 'Selected Parameters' panels, one pointing right and one pointing left.

Figure 22: Random Forest Optimize Parameters.

The parameters for our “Random Forest” operator follow a least square criterion with a maximal depth of -1. We use a maximal depth of -1 to allow our tree to grow as deep as possible to fit our data, ultimately reducing our RMSE.

 **Random Forest (2) (Random Forest)**















number of trees 	1	
criterion 	least_square	
maximal depth 	-1	
<input type="checkbox"/> apply prepruning 		
<input type="checkbox"/> random splits		
<input checked="" type="checkbox"/> guess subset ratio		
<input checked="" type="checkbox"/> use local random seed		
local random seed	2024	
<input checked="" type="checkbox"/> enable parallel execution		

Figure 23: Random Forest Parameters.

## Neural Networks

We decided to use Neural Networks as one of our models since our dataset has high dimensionality. The specific RapidMiner parameters are shown below:



 **Neural Net (2) (Neural Net)**
















hidden layers	 Edit List (2)...	
training cycles	200	
learning rate 	0.01	
momentum 	0.9	
<input type="checkbox"/> decay		
<input checked="" type="checkbox"/> shuffle		
<input checked="" type="checkbox"/> normalize		
error epsilon	1.0E-4	
<input checked="" type="checkbox"/> use local random seed		
local random seed	2024	

Figure 24: Neural Net Parameters.

For our hidden layers, the parameters are shown below:

 Edit Parameter List: hidden layers ×

 Edit Parameter List: **hidden layers**  
Describes the name and the size of all hidden layers.

hidden layer name	hidden layer sizes
anime	6
anime2	6

Figure 25: Neural Net Hidden Layers.

Our reasonings for the hidden layer sizes and number of hidden layers of 6 and 2 respectively is due to the fact that beyond these values, the accuracy of the model dips, presumably due to overfitting. For training cycles, we decided to pass it through the ‘Optimize Parameters’ operator with the following characteristics:

Min	Max	Steps	Scale
200	300	50	linear ▼

Figure 26: Neural Net Optimize Parameters

The resulting optimized training cycles after running the model is 244. The RMSE and  $R^2$  of the model is shown below:

**root\_mean\_squared\_error**

`root_mean_squared_error: 0.432 +/- 0.000`

Figure 27: Neural Net RMSE

**squared\_correlation**

`squared_correlation: 0.814`

Figure 28: Neural Net R2

Ultimately, not as good as the CART model, but still good performance.

## Challenges and Limitations

The original dataset had over 19,000 rows, which would dramatically increase model runtimes on RapidMiner. The Backwards elimination operator, in particular, the backward elimination operator seemed to take a long time to finish. As a group, we decided to make the

trade-off of reducing the dataset, possibly not capturing all the details of the data, into a sample for greater model runtime flexibility. Another limitation of the model was multicollinearity between variables, one relationship we noticed that had a high correlation was between 'ranking' and 'score', which may have skewed our conclusions. As a result of our data-cleaning processes and the general structure of the original dataset itself, the data we used to run the models had missing data that may have possibly affected the results. In the original dataset, anime that were still ongoing did not have an episode number, which resulted in the 'episodes' attribute for that anime being blank. For our data-cleaning, we decided to just outright remove records that had missing data. As previously stated, some of the ongoing animes in our dataset were the most popular and largest animes at the time of collection, so our results may have been affected.

## **Insights and Recommendations**

Based on the results of our analysis, we have a few insights for those looking for new anime recommendations. To begin, throughout our analysis, our models consistently showed that members, popularity, ranking, and episodes were the strongest predictors for score on MyAnimeList. Of note, our CART model was our best model for predicting data, with the lowest RMSE value of 0.414 and an  $R^2$  of .0.834. Meanwhile, according to our findings, genre appeared to have very little predictive value for a show's score. However, one should note that due to the subjective nature of preferences genre may still be a valuable tool for an individual to account for when looking for anime recommendations. Ultimately, when it comes to media consumption, while models can point you in the right direction, individual preferences are still the biggest factors for enjoyment, and no model will be a perfect fit.

## References

Singh, H. (2024, January 5). *Exploring the global impact on the anime market*. BCC Research

Blog. <https://blog.bccresearch.com/exploring-the-global-impact-on-the-anime-industry>