

ANIME4YOU: AN INTELLIGENT ANALYTICAL FRAMEWORK FOR ANIME RECOMMENDATION AND PERSONALIZATION USING AI AND BIG DATA ANALYSIS

Kaiho Cheung¹, Ishmael Rico², Tao Li³ and Yu Sun⁴

¹Sentinel Secondary, 1250 Chartwell Dr, West Vancouver, BC V7S 2R2

²University of California, Berkeley, CA, 94709

³Purdue University, West Lafayette, IN 47907

⁴California State Polytechnic University, Pomona, CA, 91768

ABSTRACT

In recent years the popularity of anime has steadily grown. Similar to other forms of media consumers often face a pressing issue: "What do I watch next?". In this study, we thoroughly examined the current method of solving this issue and determined that the learning curve to effectively utilize the current solution is too high. We developed a program to ensure easier answers to the issue. The program uses a Python-based machine learning algorithm from Scikit-Learn and data from My AnimeList to create an accurate model that delivers what consumers want, good recommendations [9]. We also carried out different experiments with several iterations to study the difference in accuracy when applying different factors. Through these tests, we have successfully created a reliable Support vector machine model with 57% accuracy in recommending users what to watch.

KEYWORDS

Machine learning, anime, recommendations, Python.

1. INTRODUCTION

Anime commonly refers to Japanese animated cartoons and despite common belief it is aimed at both adults and children [7]. Over the last decade the popularity of anime has significantly increased especially in North America. We are starting to see subscription streaming services such as Crunchyroll pop up. For reference the anime industry as a whole is worth around 20 billion USD and still growing. However anime watchers suffer the same problem as other forms of entertainment watchers: the dilemma of what to watch next [12]. The simple solution is to go on the internet and ask; however like any other medium, taste is subjective. We developed a solution that recommends anime to you. Using AI we can provide quality and objective recommendations [10]. This solves the issues of scrolling past pages of forums of people arguing about who has the better taste. This solution is geared towards you and can be corrected to fit your taste perfectly. We are using data from MyAnimeList (MAL) as this is by far the largest wiki/rating site currently online. What you rate in the past is automatically in the programming, significantly reducing training time. By extension this program can also be attuned to manga and light novels, to give users the full experience of this culture [8]. With the growth of these systems more people are going to be watching anime and thus driving the industry forward and creating a positive feedback loop.

For most people, the main solution to the dilemma of what to watch next is to go on the internet and search up said problem. Upon hitting enter the user is flooded with a wall of links to various internet forums and posts. Most noticeably Reddit, Reddit is one of the largest online communities and as such contains a lot of varying opinions. Take the subreddit r/AnimeSuggest, a forum dedicated to help others find anime. Most of the posts have replies with very different recommendations, actually, you even see people arguing over what they suggested. To newcomers of the anime community, this is incredibly overwhelming as they don't know who to listen to and have to base their decision on popularity or cross checking other posts. This process can take hours and can keep people away from watching anime. A similar recommendation program that implements AI is RikoNet, RikoNet uses an auto-encoder to cluster and filter data to make predictions [13]. This requires training time and user feedback and is not suitable for everyday users.

The method that we are using is building a machine learning model in Python that classifies anime similar to the user input. To do this we are using a non linear support vector machine (SVM) which classifies data points (vectors) using linear regression. This method allows us to have multiple variables of consideration which makes the predictions more accurate. SVMs are also very easy to build, come with built-in optimization, designed to work with unstructured data and scale well to high dimensional data. This makes SVMs perfect for recommending anime. We are using data from myanimelist.net which is the most well known and largest anime wiki/forum to ensure accuracy.

Experiments are performed to calculate and compare accuracy for determining the most appropriate machine learning model through the implementation of support vector learning (SVM) and regression models. In the experiment, we tested different models through adjusting the regression model, polynomial parameters, and inputted data sets.

We experimented with both the training data set size and the genre used for the SVM model to look at their effects on accuracy. In the first experiment we assessed the effects of changing the rows used for the train data [14]. For example instead of using row 0-100 as train data we would change it to row 200-300. This is to determine the optimal row for training data. The second experiment we modified the genre and examined what genres had more impact on the SVM. From this, we can determine what genre was more "important" and what can be removed. This experiment was mainly to increase performance as the more X variables we have, the more complicated the model is leading to longer compute time.

The way the paper will be structured after this introduction will be as follows: Section two is about challenges that the user face using other solutions currently available; Section three will discuss our proprietary solution and what methodology is associated with the solution; Section four details the experiments we've conducted and analyses on the data we gathered; Section five is dedicated to exploring related work in this area, as well their methodology when approaching the same problem. At the end section six is a conclusion and future development of this project.

2. CHALLENGES

Challenges for current systems have been identified as follows.

2.1. Finding out useful information

When answering the question of what to watch next most people turn towards the internet for help [15]. There are a lot of good informative recommendations out there on forums and

YouTube [11]. The problem is that there is too much information and too many factors at play for people to sort out what recommendation is good for them. The main factor that makes it impossible to decide who to listen to online is the arguments. Different users post different recommendations and often argue that their tastes are the best. A Lot of these posts are very long and consist of many back and forth requiring a lot of time to sort through. Sometimes people speak highly of an anime because the anime is infamously bad which can lead to even more confusion. Searching online can give you a general sense of what is preferable to watch but that takes time and previous experience.

2.2. Saving training time

Currently most of the anime recommended on anime sites are manually coded meaning that it is based on the tastes of the person who coded it or just based on popularity. Although there are ones that do incorporate machine learning, some examples we found include RikoNet(RKNE) which utilize “deep auto-encoders for the tasks of predicting ratings”. However thesis types are limited by their models. The model RKNE has is very similar to what Netflix has, where it takes user data, makes a prediction and then uses user feedback to improve itself. In the long run this system is incredibly effective. The downside is that it takes time to train and you need constant user feedback. So it is hard to apply to small sites and gives sub-optimal recommendations in the early stages. Our program just gives a few anime based on statistics and has zero training time making it more viable to consumers.

2.3. Limited choices

When it comes to what to watch, premium streaming services like Crunchyroll and Funimation have very well developed “manual” recommendation systems. They are big enough to hire professional people to recommend what to watch next and for the most part, they are on point. However, anime licensing is incredibly expensive and can cost millions of dollars. For example Neon Genesis Evangelion, a very famous anime, will not be recommended on any of the aforementioned sites because Evangelion is licensed by Netflix. This makes Crunchyroll and Funimation recommendations undesirable as they can only recommend what they have. The alternative is to go to MAL -which has all the anime- but that leads to the first challenge: Over complicated reviews and user arguments. Our system has the best of both worlds where it is simple to navigate and has all the possible anime.

3. SOLUTION

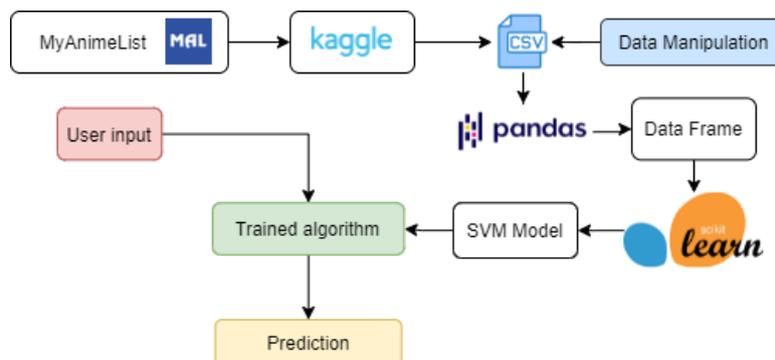


Figure 1. Overview of the project

To create this program we first had to collect data to train the machine learning algorithm. We decided to collect data from a site called My Anime List (MAL) because it is the largest databases and communities site for anime, manga and light novels. From this data, the machine learning algorithm will fit the data and can create a regression curve. The user will be asked to input whether they like a genre or not. From the user's answer, a point will be created on the regression curve and that will be used to predict what anime to recommend.

Originally we wanted to get the information directly off of MAL using their built-in API calls however that overreach the scale of this project. Instead, we got MAL data from Kaggle, a public database. We downloaded a CSV file that included anime_id, name, genre type, episodes, rating and number of members. However, this data was not usable as the genre types were not isolated in individual columns but instead a cluster of strings. Pandas the data analysis and sorter does not take strings so a work around was needed.

anime_id	name	genre	type	episodes	rating	members
32281	Kimi no Na	Drama, Romance, School, Supernatural	Movie	1	9.37	200630

Figure 2. CSV file

We had to reformat the data so that genre was isolated and had an integer value attached such as the picture below [6]. When we reformatting the data we had to create multiple columns so excel could check each time if it had the string it was looking for. For example it would check if the anime had a romance tag it would put a 1 and everything else would be 0. The data would look like the example below.

Drama	Romance	School	Supernature
1	1	1	1

Figure 3. Data example

This way we can actually feed these numeric values into Pandas to create data frames. The data frames are then used to train our model. For this project we decided to use Scikit-learn which is a python based machine learning algorithm and more specifically using Support Vector Machines to do classification. Our X axis is the genre and Y is the titles. From there we get a set of user inputs which will define a specific parameter and using linear regression predict or in this context suggest an anime to the user.

```

1 import pandas as pd
2 from sklearn.model_selection import train_test_split
3 from sklearn import svm
4

```

Figure 4. Code line 1-3

Lines 1-3 Imports the necessary code libraries to function. pandas is the data sorter and sklearn is the SVM machine learning code.

```

6 df = pd.read_csv("CleanedAnime.csv").dropna().iloc[:100,]
7 X = df[['Action', 'Adventure', 'Cars', 'Comedy', 'Dementia', 'Demons',
8       'Drama', 'Ecchi', 'Fantasy', 'Game', 'Historical', 'Horror', 'Josei',
9       'Kids', 'Magic', 'Martial Arts', 'Mecha', 'Military', 'Music',
10      'Mystery']]
11 Y = df[['Title']].index.tolist()
12
13 #X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = .3, random_state=0)
14
15 model = svm.SVC()
16 model.fit(X,Y)
17 print(model.score(X,Y))

```

Figure 5. Code line 6-17

Lines 6-11 is Setting up the training data location from pandas (pd) (CleanedAnime.csv) and the training set size. Secondly it defines what the X and Y variables are for the SVM. As seen above the X includes all the genres an anime can have and the Y is the title.

Lines 15-17 is creating a SVM, testing it on our data and then printing out the score.

```

19 df = pd.read_csv("CleanedAnime.csv").dropna().iloc[100:200,]
33 df = pd.read_csv("CleanedAnime.csv").dropna().iloc[200:300,]

```

Figure 6. Code line 19 and 33

Lines 19-45 follow the same code showing lines 6-11 except for the training data set is changed to row 100 to 200 and row 200 to 300 in the CVS. show in Lines 19 and 33.

```

48 df = pd.read_csv("CleanedAnime.csv").dropna().iloc[:100,]
49 X = df[['Action', 'Drama', 'Comedy', 'Horror']]
50 Y = df[['Title']].index.tolist()
51
52 #X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = .3, random_state=0)
53
54 model = svm.SVC()
55 model.fit(X,Y)
56 print(model.score(X,Y))
--

```

Figure 7. Code line 48-56

Line 48-56 set the training set to the first 100 rows. However we changed the genres used in the X axis. Then in line 54-56 we test the accuracy of the data.

```

59 X = df[['Dementia', 'Demons', 'Drama', 'Ecchi', 'Fantasy', 'Game']]
69 X = df[['Historical', 'Horror', 'Josei', 'Kids', 'Magic', 'Martial Arts', 'Mecha', 'Military',
'Music', 'Mystery']]

```

Figure 8. Code line 59 and 69

Line 56-76 follow the same code showing lines 48-56 except for the different genres shown in Lines 59 and 69 aside from that the rest performs the same tasks.

4. EXPERIMENT

4.1. Experiment one

In experiment one as previously mentioned we wanted to determine the effect of specific rows used as training data and the accuracy of the SVM model associated with those rows. We performed a series of tests with and limited the test data to 100. The reason why we choose 100 rows as the size is because we are testing accuracy by checking how similar our model's prediction is compared to the validation set (a set of data with known characteristics) however we do not have a validation set. We simply check the similarity between the predicted and the whole data set. This means that the larger the training set the more accurate it is going to be because the SVM would be based exactly on the test set [5]. This is undesirable because like a student studying with the actual test question the algorithm isn't developing anything, instead it would already know the answer when challenged with the real data. We found that 100 rows was not enough to trigger the aforementioned problem as our data set is about 11300 rows long, and that it yielded realistic results. Our data are sorted by popularity and score meaning that "better"/more desirable anime was at the top and less known/ lower score ones were on the bottom. On to the test itself we wanted to see if using high ranking shows would affect accuracy so we set the training data rows to 0 to 100, then 100-200 and finally 200-300 and here are the results. Surprisingly when using the top 100 anime in training the accuracy is less compared to the training with less popular anime. Rows 100-200 and Rows 200-300 yielded the same results. The reason why this might happen is because popular anime often come with sequels effectively doubling the effect on the model thus making it worse facing the full data set. Further research is needed.

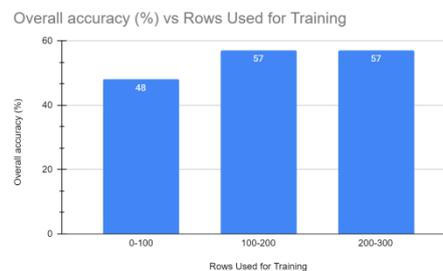


Figure 9. Overall accuracy VS rows used for training

4.2. Experiment 2

In experiment 2 as previously mentioned we wanted to determine the effect of specific genre on the accuracy. Initially we designed this experiment because we suspect that some genres are more of a key indicator than other genres. By knowing what is key we can reduce the number of x variables thus increasing performance subsequently seaming resources. We believe that common genres are better because they cover more of the data set while distinct genres will yield poorer results due not enough data. Our theory is based on the way SVM operates. More common genres would create support vectors that nudge the regression line in a more general direction thus covering more and leading to high accuracy. There is the chance that because of the commonality the hypnosis line covers what they should. On the other hand more niche genres would perform worse because of their lack of data points [4]. Also more niche anime would have more niche genres and as a support vector is based on niche anime it would not accurately predict the rest of the population. For the purpose of testing the genres we set the training data to the first 100 rows. The first iteration is what we thought is the most common genre which includes: Action, Drama,

Comedy, Horror. The second iteration is a mix of both common : Drama, Ecchi, Fantasy and uncommon: Dementia, Demons, Game. The third iteration has all the uncommon tags: Historical, Horror, Josei, Kids, Magic, Martial Arts, Mecha, Military, Music, Mystery. The result was unexpected. If we look at the experiment 's first iteration which scored 48% in accuracy of courses by remaining genres we expected the accuracy to drop but not by this much. The accuracy of all three iterations did not reach half the accuracy of the one with all the genres. Another surprising result was that the one with all the uncommon tags scored the highest with the mix scoring the lowest. But based on the difference the one with all common tags performed decently. After this experiment we think the perforce increase in optimizing the genre is simply not due to the heavy accuracy penalties and our hypothesis is partly true.

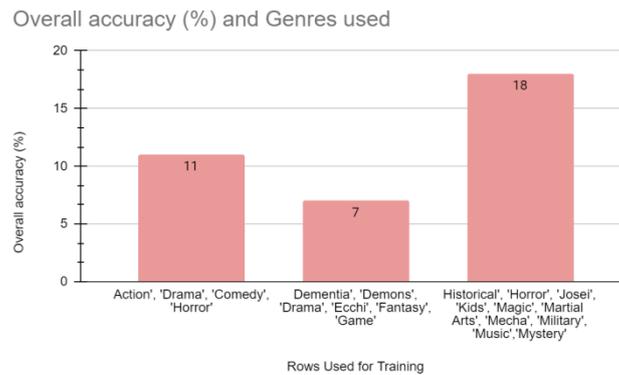


Figure 10. Overall accuracy and genres used

5. RELATED WORK

Mangaki is a person who recommends that quickly profiles users of what precedence they have. It uses a series of anime to identify what the user likes with buttons Like, Dislike, Neutral, Will see, Wont see. Then after the initial set of anime they would start recommending anime based on similarity. This is similar to our system to recommend based on similarity but it uses a filter with a formula instead of algorithm. This site is easy to use but it requires the user to have watched a few anime before and especially the initial set “classics” to actually work.

AniReco is a visual based anime person who recommends that finds other anime with similar tags. These tags include genre, voice actuator, director etc. they then connect anime with lines and the closer an anime is the more similarities it has. AniReco has a similar approach of finding similarity where we differ is the algorithm used. They use a vector space model while we used a support vector machine. Another difference is that AniReco requires the user to “train” the filter first by giving ratings of shows they watched. while ours can just be used base on preferred genres.

RikoNet is a person who recommends both a novel and Nike but for the purpose of this paper we will only look at the anime recommendation. The person who recommends let the user put in five anime they liked from there it creates a profile. This profile is then put through a hybrid filter and from the database it gives the user five recommended anime. Like Mangaki this requires the user to have watched previous shows or else it will just give out five most popular anime. One downside is that RikoNet is an engine so individual consumers have a hard time accessing it as it was designed for websites.

6. CONCLUSIONS

We created a machine learning algorithm to classify anime to ensure that algorithm is optimized and identify possible factors that can affect it [1]. We carried out two experiments. In the first experiment we decided to see if popularity within the training set affected overall accuracy. We found that using the top 100 anime acquired a lower accuracy and we suspect that is due to popular anime squalls and the nature of SVM [2]. In our second experiment we carried trials to determine whether common and uncommon genres had an effect on accuracy and the results showed that having uncommon genres will yield better accuracy. However it is best to use all the genres when modeling as the perforce gain is not worth the 62.5% drop in accuracy. In conclusion, our SVM model with training data set rows from 100-200 and all the genres delivers excellent performance (52% accuracy) when classifying an anime. As such we believe this is a reliable anime recommended for consumers.

The current limitation of the program comes from the data set we are currently using. Due to the not using API calls, the current data we are using is about one years old which can cause issues regarding accuracy. In terms of accuracy the program scored very high however it was tested on the data it trained on. The validation data set was also derived from the main data, this can lead to questionable accuracy results if new data is introduced. The biggest limitation is taste, whether an anime is good or not is highly subjective, using data to predict something subjective is sub-optimal.

A stretch goal of this project would be MAL profile integration [3]. This means that we would collect data from the user's MAL profile and analyze what anime they have watched. Then train with that data to better suit them. Another thing to implement is a score and member count factors into the program where it would favour high ranking and more popular shows to improve end user experiences.

REFERENCES

- [1] Anime and Manga Database and Community. MyAnimeList.net. (n.d.). Retrieved October 10, 2021, from <https://myanimelist.net/>.
- [2] R/animesuggest. reddit. (n.d.). Retrieved October 10, 2021, from <https://www.reddit.com/r/Animesuggest/>.
- [3] Hernández, Á. D. H. (2018, September). The anime industry, networks of participation, and environments for the management of content in Japan. In *Arts* (Vol. 7, No. 3, p. 42). Multidisciplinary Digital Publishing Institute.
- [4] Ota, S., Kawata, H., Muta, M., Masuko, S., & Hoshino, J. I. (2017, September). AniReco: Japanese Anime Recommendation System. In *International Conference on Entertainment Computing* (pp. 400-403). Springer, Cham.
- [5] Soni, B., Thakuria, D., Nath, N., Das, N., & Boro, B. (2021). RikoNet: A Novel Anime Recommendation Engine. *arXiv preprint arXiv:2106.12970*.
- [6] Vie, J. J., Laily, C., & Pichereau, S. (2015). Mangaki: an anime/manga recommender system with fast preference elicitation. *Tech. Rep.*
- [7] Clements, Jonathan. *Anime: A history*. Bloomsbury Publishing, 2017.
- [8] Brenner, Robin E. *Understanding manga and anime*. Greenwood Publishing Group, 2007. [9] Epstein, Joshua M. "Why model?." *Journal of artificial societies and social simulation* 11.4 (2008): 12. [10] Cames, Kathleen, and Administrative Analyst II-Grants. "Recommendation." *City* (2006).
- [11] Shani, Guy, and Asela Gunawardana. "Evaluating recommendation systems." *Recommender systems handbook*. Springer, Boston, MA, 2011. 257-297.
- [12] Dyer, Richard. *Only entertainment*. Routledge, 2005.
- [13] Soni, Badal, et al. "RikoNet: A Novel Anime Recommendation Engine." *arXiv preprint arXiv:2106.12970* (2021).

- [14] Vanschoren, Joaquin, et al. "Experiment databases." *Machine Learning* 87.2 (2012): 127-158. [15] Gralla, Preston. *How the Internet works*. Que Publishing, 1998.

AUTHORS

Kaiho is a student currently attending Sentinel Secondary in West Vancouver. He is interested in pursuing computer science and engineering in the future.



© 2021 By AIRCC Publishing Corporation. This article is published under the Creative Commons Attribution (CC BY) license.