ORIE 4741 Project Midterm Report

Ben Rosenberg (bar94), Brennan O'Connor (bo92), Cooper McGuire (cjm424), Zach Katz (zdk4)

November 1, 2021

Description of data

The data we're using in our project come from MyAnimeList.net, a site on which people can upload their lists of watched anime (animated, typically Japanese, TV shows and movies). MyAnimeList (henceforth referred to as MAL) is similar to iMDB: users can rate anime (1-10), leave reviews, and interact with other users in forums. In our analysis, we use a Kaggle dataset composed of data scraped from MAL regarding users and the anime they've watched.

Descriptions of datasets

The Kaggle dataset is composed of several files, some of which are "cleaned" versions of others. A table of files and descriptions is as follows:

file	description
`AnimeList.csv`	raw list of anime on MyAnimeList
`UserAnimeList.csv`	collection of the individual anime that each user on the site has watched
`UserList.csv`	list of users on the MyAnimeList site
`anime_filtered.csv`	filtered version of `AnimeList.csv`
`anime_cleaned.csv`	cleaned version of `anime_filtered.csv`
`animelists_filtered.csv`	filtered version of `UserAnimeList.csv`
`animelists_cleaned.csv`	cleaned version of `animelists_filtered.csv`
`users_filtered.csv`	filtered version of `UserList.csv`
`users_cleaned.csv`	cleaned version of `users_filtered.csv`

The filtered versions of the files contain only those users with birth date, location, and gender filled. The cleaned versions of the filtered files have outliers removed (e.g., users with excessively large watch times), and nonsensical data removed (such as users with episode counts that exceed the episode count of a given anime). Of these files, we're going to be using the cleaned ones. The missing data that was removed in the cleaning process is not necessary in order to ensure that our analysis is done with sufficient rows; since we started with 80 million users, the paring-down of the usercount left us with enough data.

Location data transformation

Since we want to analyze user location as one of our features, we needed to parse the locations given to us by users. Unfortunately, MAL has no standardized way for users to input their locations; instead, users simply type in whatever they want. This led to our having to discard a large quantity of data due to poorly-defined locations.

Furthermore, we decided (for the sake of simplicity) to focus our analysis solely on the users whose location text implied that they were in the United States, with the goal of analyzing location on a state-by-state basis. To do this, we made several dictionaries of locations, and used these dictionaries to parse the location strings given by the users. We created a string-parsing function using these dictionaries and then applied it to the relevant dataframe columns. Once this was done, we were able to isolate the rows which had location data that was convertable to a US state, and put that data into a separate dataframe. Finally, we converted said dataframe back into a CSV to get the final version of the `users_cleaned.csv` dataset.

Initial regression

The first question we sought to ask in our project is, "Can we predict how much anime someone has watched?" For this regression, we will only be using the `users_cleaned` dataset with the initial feature engineering described earlier.

Feature selection

Here is a table of the covariates in the regression we'll be performing:

covariate	description
average rating	mean of the user's ratings of anime on their anime list
join year	year that the user created an account on MAL

covariate	description
age	age of the user, as defined by their date of birth
year last online	last year the user was online
gender (M, F, NB)	one-hot: user's gender (either male, female, or nonbinary)
location (state)	one-hot: user's location (2-letter abbreviation of the user's location (state) in the US)

We used these features because we think they will be useful for the regression. Age, gender, and location are demographics that we want to test, while average rating can be seen as a proxy for the critical nature of a viewer. Join year and year last online are best thought of in conjunction as a proxy for activity in the anime sphere. For example, someone who joined a long time ago but was online recently would be considered to be very active, while someone who has joined recently and is inactive would be considered to be very inactive.

Using all of these features will extract as much information as we can from our dataset, and will therefore help us to avoid underfitting. With this many features, overfitting is certainly a concern, but we will be using various techniques to avoid overfitting which we'll discuss in the next sections.

For starters, to evaluate each regression technique, we divided the dataset using an 80:20 train:test split.

Linear regression

Our first approach was to use simple linear regression. Unfortunately, our model turned out to be rather unstable. The intercept was 2.08×10^{12}, and many of the coefficients were on the order of 10^{10}. Additionally, we suspected that some of the features were collinear because of the large, unstable coefficients. We tried normalizing the features and removing extreme label outliers, but this did not improve the results. Note that we also tried normalizing and removing outliers on future regressions, but it didn't improve results, so we kept the original data.

Lasso regression

In order to stabilize the model, we modified our regression model to include the ℓ_1 (Lasso) regularizer:

$$\underset{w}{\operatorname{argmin}} \|y - Xw\|^2 + \lambda \|w\|_1$$

As we can see, we need to select the regularization parameter λ. In order to wisely choose this parameter (which will avoid overfitting), we used 5-fold cross-validation and tested 100 possible values of λ from 0.001 to 10. The optimal parameter, which we ended up using in the actual regression, was 0.101.

The advantage of the Lasso is that it chooses a model that is both stable and sparse. Here are tables showing the *nonzero* weights selected by the regression:

coefficient	weight	coefficient	weight
average rating	-3.396	gender (♂)	11.446
join year	-4.829	location (CA)	3.994
age	0.735	location (NY)	2.079
year last online	8.642	location (OH)	-2.536
gender (♀)	-3.622	location (TX)	0.958

These results indicate that men are more likely to have more days of anime views, and women are more likely to have fewer anime days. (The regression didn't have enough data points to infer viewership patterns about nonbinary individuals.) We also see that users who were recently online and users who joined earlier are more likely to have more anime days. Finally, users who had a *lower* average rating (and were more critical of anime in general) were more likely to have more anime hours. The one-hot locations that were selected did contribute to the model's predictive ability, but since these one-hots are inherently collinear, it is unfortunately difficult to make inferences about them.

Importantly, the model did not overfit; the training MSE was 2,238 and the testing MSE was 2,074. Because of scaling, the MSE's are difficult to interpret on their own. Instead, we can use the coefficient of determination $\mathcal{R}^2$ to analyze the effectiveness of this model. The coefficient of determination is a measurement of the proportion of variation in the dependent variable that can be measured by the covariates[1]. The $\mathcal{R}^2$ for the training data was 0.255, and the $\mathcal{R}^2$ for the testing data was 0.257.

Although these $\mathcal{R}^2$ values may seem low, viewership data is quite noisy, so we were satisfied with this result since our model does give a general sense of whether a given person is more likely to have watched more anime. This will be discussed more in the takeaways section.

[1] Wikpedia: https://en.wikipedia.org/wiki/Coefficient_of_determination

Random forest

In an attempt to improve the accuracy of our model, we decided to fit a random forest regressor on our training data. Random forest regressors have many hyperparameters to choose from, but of the most concern to us were the number of trees in the forest and the maximum depth of each tree. By default, sklearn's random forest implementation allows 100 trees per forest and unlimited depth of each tree! After an initial trial, these defaults led to massive overfitting. To fix this, we once again ran a 5-fold randomized CV (with 100 iterations) to choose possible parameter values between 1-50 for the number of trees, and 1-10 for the maximum depth of each tree. The optimal parameters chosen were 40 estimators with a max depth of 6. Figure 2 below shows the top 6 feature importances from this regression. These are relatively in line with the features chosen by the Lasso regression.

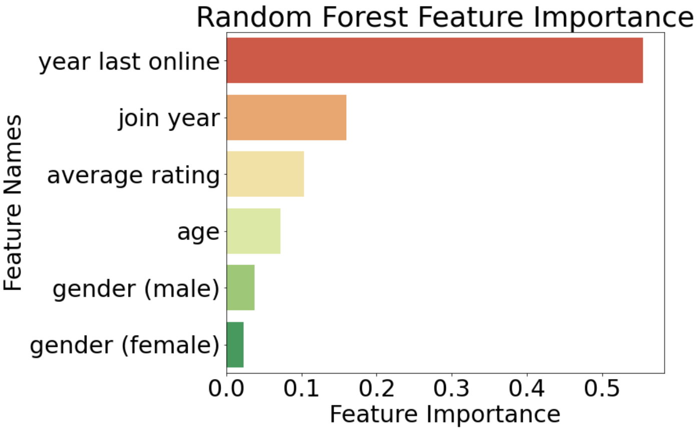

Figure 1: Feature importance from random forest. Note that locations were not included; on the whole, they were not defined as important by random forest and, being one-hots, were colinear anyway.

The training MSE was 1,984 and the testing MSE was 1,967. This small difference indicates that we chose parameters wisely to avoid overfitting.

The training $\mathcal{R}^2$ was 0.339, and the testing $\mathcal{R}^2$ was 0.296. Therefore, the random forest model was able to provide slightly better predictive power than the lasso regression, as evidenced by slightly lower MSE scores, and more importantly, the higher coefficients of determination.

Regression takeaways

Although we couldn't construct any perfect models, both the ridge regression and random forest models provide useful insight into anime viewership. If we needed to predict the number of days someone has watched anime, the random forest model is more useful since it has a higher $\mathcal{R}^2$ and lower MSE's. However, for the purposes of inference, we would turn to the Lasso regression, since we know it only chose the *most important* features. One example of such an inference would be observing that if we had a male who tended to rate anime poorly, we would expect him to have watched a substantial quantity of anime. This type of inference could be useful for an anime studio targeting an audience for a new program.

Future research

While a majority of the effort associated with data cleaning, as well as a basic initial regression, have been accomplished, outstanding are several core questions to be answered. We aim to inform management at various U.S. anime distribution levels of the preferences of their current (or future) audience. Potential future research questions include:

1. **Are there distinct clusters of anime-watchers?** Through demographic and usage data tied to every user, we aim to implement a form of k-means clustering to partition the user set. This will inform anime studios of stereotypical user profiles from which to infer preferences. Clustering will also provide insight into "lookalike" audiences for the company's user base, giving marketing management concrete audiences to prioritize. Clustering may also facilitate more effective budget allocations for marketing and acquisition of anime streaming/broadcasting rights.
2. **Can we effectively predict the favorite genre/genres of a given user?** Such a model could help to inform an anime recommendation system that suggests certain genres or shows to certain users, and thus increases one's interest and likely retention on a given site (e.g., CrunchyRoll, Netflix). Furthermore, companies with a hand in production or providing subtitling would know which types of shows to develop or work on, based on the target audience.
3. **Can we effectively predict the rating of a given anime by a given user?** If management were trying to target a specific demographic, for example, they can see if the anime in question would likely be highly rated by that group. More broadly, since we include the title as a feature in the model, a studio can experiment with different potential titles for an anime and see how the ratings vary.