Predicting User Ratings for Anime-Style Images

Alexander Lamson
College of Information and Computer Sciences
University of Massachusetts Amherst
Amherst, MA 01003
alamson@umass.edu

Abstract

The purpose of this project is to predict how users will rate given anime-style images. Generally, images with higher scores are regarded as being of higher quality and more liked by the community. By creating a regression function which predicts the score of an image, the quality of arbitrary novel images can be approximated.

1 Introduction

Safebooru (http://safebooru.org/) is a tag-based image archive maintained by anime enthusiasts. It allows users to post images and add tags, annotations, translations and comments. It's derived from Danbooru, and differs from it in that it disallows explicit content. It's quite popular, and there are more than 1.8 million images as of November 1, 2016.

Each image comes with some metadata, which includes tags, an ID, a post timestamp, the image dimensions, a URL for the image's source, the rating (on Danbooru this could be safe/questionable/explicit, but on Safebooru it's always safe) and a simple integer score which each user can vote up or down. As is generally the case in online forums and social media, the score is an aproximation as to how much the community "likes" an image.

The goal of this project is to create a function which takes in the the tags of any arbitrary image as input and outputs the predicted score for that image, with the predicted score being as close to reality as possible. As machine learning problems go, this is a regression problem. This function could potentially be used to find under-rated images with low scores that actually are high quality. It could also be used to quickly check how popular an image will be as soon as its posted.

It was found that image scores could be predicted with a mean absolute error of 0.75378 points. The best method found was to decrease the number of features from 1,000 to 100 using PCA then to use those features to train an ordinary least squares linear regressor.

2 Data Set

2.1 Tags

Figure 1 contains a set of example image results found by searching for the tag *playing_instrument*. Associated tags can be found on the left column of the screenshot. Some examples of tags are *1girl*, *aqua_eyes* and *blue_necktie*. The number next to each tag is the number of times it has ever been used to tag an image. "solo" is the most common tag at 782,924 occurrences.

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

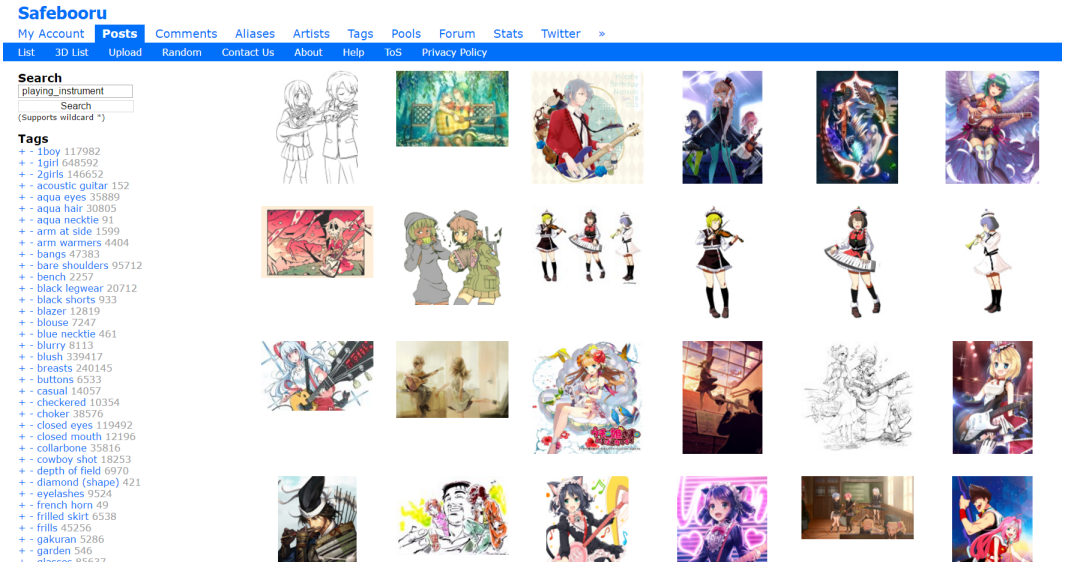

Figure 1: An example query showing images with the label *playing_instrument*.

2.2 Scores

The distribution of scores is not uniformly distributed among the images. As shown in Figure 2, the bulk of images have scores of 0 or 1. Only 28.9% of images have a score greater than 0. This poses a practical problem of how to create an accurate regressor that doesn't return 0 too frequently.

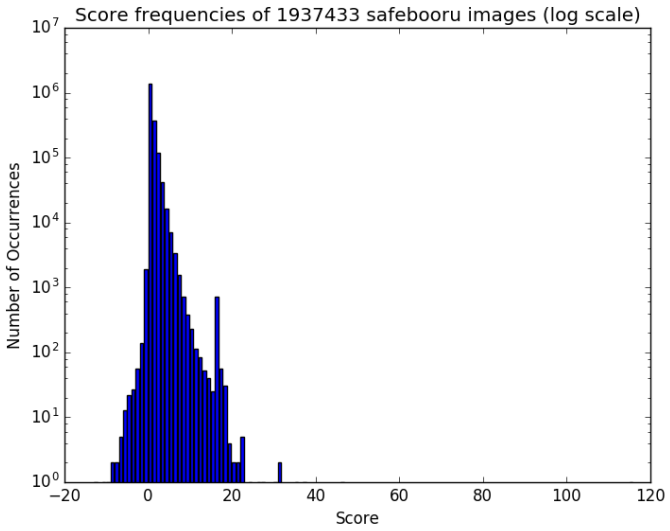

Figure 2: A histogram showing the distribution of scores among the sampled images. Note the logarithmic scale.

3 Proposed Solution

First, all the images were downloaded via Safebooru's API. It took 5 hours and 12 minutes to download the metadata for all 1,937,467 images. Because the quantity of data was so large, sampling was required to train the models in a feasible amount of time. This was done by sampling a subset of the images. All images with scores less than 0 or greater than 1 were sampled. Images with scores of

0 or 1 were sampled probabilistically such that 20,000 images were expected to be sampled for each score. This process resulted in 583,291 samples.

Next, a subset of tags were selected as features. All tags were sorted by their frequency in the population. The *tagme* tag was disregarded, because images containing that tag tended to have very few tags in general which would have made them poor samples. Tags that appeared in less than 100 images were also disregarded. Out of the remaining tags, the top 1,000 were selected.

After the tags were selected, a function was created which mapped from arrays of tag names to binary vectors, with "true" representing the presense of a tag and "false" representing an absence. The data was vectorized by this function and subsequently separated into training, testing and validation sets.

The data was then passed to two regressors. In the first one, PCA was applied to reduce the number of features to 100. Then ordinary linear regression was performed, resulting in a mean absolute error of 0.753783070014 points. In the second regressor, the features were passed directory to a decision tree regressor. The tree had a max depth of 4, required at least 10,000 samples per split, and at least 5,000 samples per leaf. This resulted in a mean absolute error of 0.767630618964 points. Note that as these error rate are not squared, they map directly to the point system. Reviewing Figure 2, it's clear that score range from about -10 to 20 (a spread of more than 30 points). The training, testing and validation sets were broken in to 80/10/10 sizes, respectively.

4 Experiments and Results

4.1 Change in Sampling Procedure

Sampling was intitally done by taking every n-th line. This didn't work as images with score of 0 and 1 occured so frequently that the regressor began to overfit on the few high-scoring examples that were sampled.

4.2 Graphing the Regression Tree

The decision tree regressor descibe in section 3 is visualized in Figure 4. The tag *original* denotes when the artwork is an original creation and does not contain characters from established anime or game. The tag *kantai_collection* refers to a popular online game which is the target of much fan-made art.

5 Conclusion

There is much area for research remaining, in both the machine learning and data mining aspects of this dataset. Machine learning could be used to asociate visual image features with respective tags. Inter-tag correlations could also be measured exhaustively to further optimize the feature selection step. When viewing Figure 2, there is a noticable spike in frequency of scores 16. It would be interesting to try to discover why that is.

Interestingly, the tag *1boy* has 118,897 occurrences while *1girl* has 636,296 occurrences. There's obvious disparity between how often males and females occur in the images. It would be interesting to try to find out why that is, as well as what tags are most correlated with males as well as females.

6 Related Work

This is the first is instance this author is aware of this dataset being used for regression scores from tags. However, this dataset has been used for machine learning tasks at least twice before.

The first instance of this usage is the program Waifu2x, which performs image super resolution to increase the size of anime-style art plausibly with minimal artifacts.

The second instance of this dataset being used is in the program chainer-DCGAN, which used a generative adversarial neural network trained on anime faces extracted from the images to generate new anime faces given a set of tag weights.

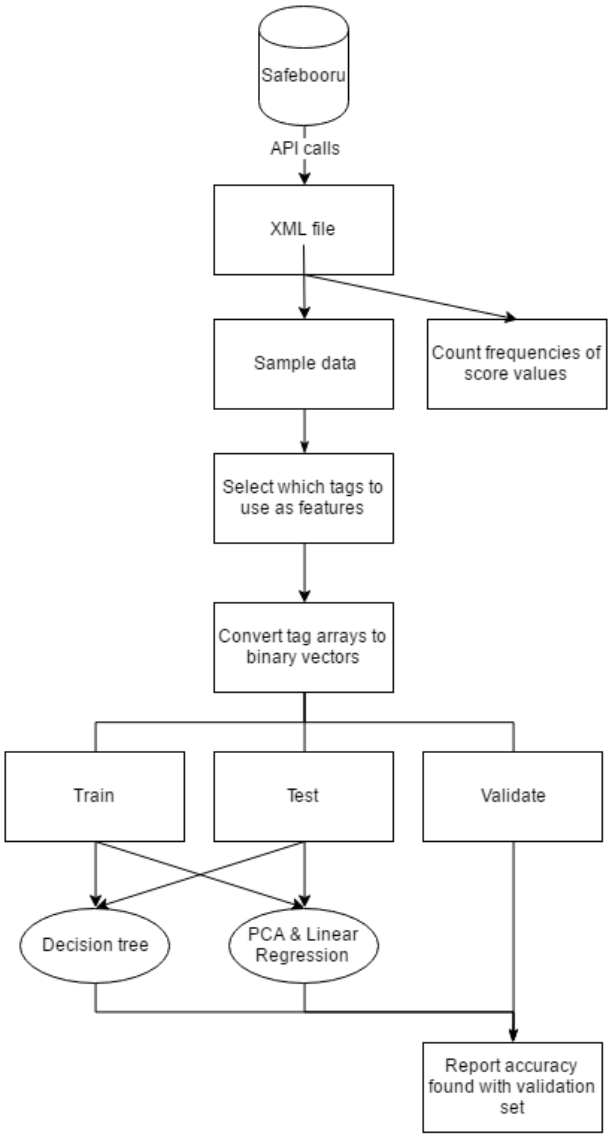

Figure 3: A visualization of the regression pipeline.

References

[1] Eiichi Matsumoto, *Chainer implementation of Deep Convolutional Generative Adversarial Network*, (December 2015), GitHub repository, `https://github.com/mattya/chainer-DCGAN`

[2] nagadomi, *Image Super-Resolution for Anime-Style Art*, (May 2015), GitHub repository, `https://github.com/nagadomi/waifu2x`

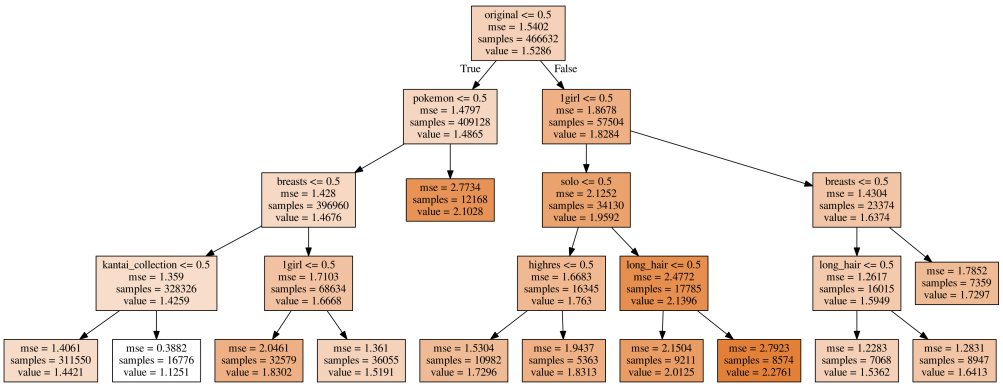

Figure 4: A visualization of the regression tree. Taking the left edge means the above tag is not present, while taking the right edge means the above tag is present.