Will Gleave (solo member)

Final Project Report

**1. Identify the Problem:**

Currently, most video game reviews are scored, but there are many reviews which only contain text detailing the reviewer’s opinion, without an accompanying score. This is most likely by design of the reviewers, as reviewers may believe that since their opinion is nuanced, it is better for a reader to read the full text to understand the entirety of the opinion, without being influenced by a single score. No metric exists to describe this situation, but this reality can be evidenced in the fact that reviews from top video game websites such as Polygon and Kotaku do not score reviews. Some general news websites which review games, such as the Washington Post, also do not score reviews.

The desired alternative solution is a model that will estimate scores for these reviews, based primarily on the textual content of the reviews. This alternative situation will allow for video game fans to see a quick summary of what a reviewer likely thinks of the game, if the fan does not have the time to read the whole review.

This problem is important to any business or entity that wants to understand the reviews of the products or services. A website such as Rotten Tomatoes, Metacritic, or any review aggregation service would care directly about assigning a score to review text. These websites synthesize many professional reviews, so an automated method for scoring these “unscored” reviews would be very useful. Further, any platform which allows reviews or feedback without a score could use a model like this to get a better estimate for the intended sentiment of the text. This could help businesses better quantify communications from customers.

Moving to the desired alternative state would of course help video game players better understand unscored reviews. More generally, it would also make it easier to quantify feedback.

**2. Define Objectives and Metrics:**

In order to transition to the desired alternative solution, my objective is to predict the review score for a game review. Since the reviews in my dataset are scored continuously from 0-10, the goal is to minimize the error between the true review score and the predicted review score.

Root mean squared error is the error metric in this project.

**3. Understand the State-of-the-Art**

Others have attempted to solve similar problems. Gupta, Fabbrizio, and Haffner predicted different restaurant quality categories (food, service, etc) based on the text of restaurant reviews [1]. Wang used text features to predict the sentiment of Yelp reviews using perceptron algorithm, Naïve Bayes, and SVM [2]. Results were mixed, with test set error rates ranging from 22% to 46%, depending on the algorithm and data preparation methods used. More generally, the field of sentiment analysis has been widely researched. This problem remains difficult to solve because of the flexibility and ambiguity of human language. Depending on the context, words can take on varying meaning, making it difficult to gauge sentiment without understanding the semantic meaning of text.

**4. Define Hypothesis and Approach**

Hypothesis: A model using TF-IDF as text features and applied machine learning will more accurately predict video game scores compared to using the mean score across all games as the predicted value.

Video game reviews from the website IGN.com make up the dataset for this project. I plan to scrape the website for review text and score data. After removing duplicates, the dataset has about 2,900 reviews. IGN is a video game website that has thousands of reviews going back several decades. This dataset may be biased towards the IGN style of writing, but this may make it easier for the model to make predictions for this dataset but make it more difficult when generalized to new data. The fact that many different authors have written the reviews help to mitigate this.

I plan to apply some of the text mining methods we learned in class for feature extraction. I plan to experiment with TF-IDF as well as topic modeling. I want to try several different models to see which type of algorithm works best, including a tree-based approach and a linear approach. These methods assume that I will have cleaned textual data as the input, and a continuous variable as the output. The dataset I am using for this problem will most likely be novel, as I will have to use web scraping to acquire it. I have not seen any previous attempt to predict video game review scores, so the TF-IDF for feature generation as well as modeling approaches should be novel when applied to this specific domain.

I will use root mean squared error (RMSE) to measure the success of this work. A low RMSE will mean that using TF-IDF for score prediction will be successful, while a high RMSE will mean that using TF-IDF will be unsuccessful. My setup will help me measure success for this project, although it will not indicate how well this will generalize to other review datasets.

**5. Execute Approach and Report Results**

After building a number of models and conducting analysis, I have learned that while the problem remains difficult, it is possible to use text features to build a model that predicts a review score better than using the average review score as the prediction. The model with the best performance was created using a TF-IDF of the corpus as the features. The input to the TF-IDF was a dictionary built from the corpus. I used a lasso regression to build the model and used cross validation to compare model performance. The top performing model had a test set RMSE of 0.977, which was a 35% improvement over the baseline score (using the average score of the dataset as the predictor). I have also learned that feature selection was more important to model performance than choice of algorithm. The difference between algorithms only appeared to be a few percentage points, while the large gains in performance came from using the correct features in the correct format. This project also reinforced the difficulty of the problem in general. While the median RMSE of .605 was pretty good, the model did a poor job with some predictions evidenced by prediction errors higher than 4.

I experimented with a number of different methods to see what gave me the best cross validation results. In addition to using the word dictionary as the input to the TF-IDF, I experimented with using n-grams as the input. N-grams typically produced worse performance, with RMSE in the range of 1.1-1.2. I also experimented using topic modeling as the model input. Despite trying different alpha values and topic numbers to tune the LDA, the LDA did not choose topics that differentiated based on review score. The LDA features were not useful for prediction. I also attempted to incorporate the genre of the game in my predictions, including a model based on genre alone, without any text data. These predictions were worse than the TF-IDF, even though I could see some differentiation based on genre in my data exploration. I tried four different methods for modeling building, incorporating both linear and non-linear methods. From one of my runs, here is a breakdown of how the different algorithms compared:

|  |  |
| --- | --- |
| Method | Test RMSE |
| Lasso Regression | 0.976 |
| OLS | 0.982 |
| Deep Learning | 0.988 |
| XGBoost | 1.058 |
| Predict Average Score (baseline) | 1.505 |

Other random train/test splits produced an ordering of results similar to this run. Note that a more extensive hyperparameter tuning process could be used for the XGBoost model and the deep learning model, as the models used for this project were vanilla. Average review scores increased by time, and also differed across game platforms. While these elements could have been incorporated into the model, the goal of this project was to focus on features derived from text. As evidenced by the results above, a model based on TF-IDF features does provide a performance boost compared to a baseline of using the average score for prediction. Generally, the model performed well, but was hurt by predictions with large errors that skewed the RMSE upwards. A median difference between prediction and actual score of .605 was calculated. Large improvements could be made to the RMSE if the predictions with largest errors could be reduced.

The data appears to be well suited for a TF-IDF to derive meaning from the corpus. There are enough reviews, and the reviews are individually long enough to give the model enough information to make predictions. It would be interesting to see if more data would improve predictive performance further. None of the other features that I scraped from the website seemed to have predictive value. While they were not the focus of this project and not extensively explored, they had limited ability to improve on the baseline score. I attempted to combine the TF-IDF with the genre feature to improve performance, but was unsuccessful. Perhaps combining some of these additional features with the TF-IDF could provide incremental performance improvements.

I used a validation set to evaluate performance. This validation set was split from the original training set, and was made up of reviews from the same website. It would be interesting to evaluate this model on reviews from other websites to see how well this model would generalize. RMSE seems to be a good metric, as it illustrates how far away on average is a predicted value from its true value.

The results of this project support my hypothesis that “A model using TF-IDF as text features and applied machine learning will more accurately predict video game scores compared to using the mean score across all games as the predicted value”. The models built using different machine learning algorithms all outperformed the baseline score. The hypothesis may not seem a difficult bar to clear, but a model with TF-IDF features also clearly outperformed models built using the genre as predictor, or models with topic models as features. There is predictive value in the TF-IDF representation of data.

This works helps us move towards the alternative scenario because it has explored several options for numerically scoring a video game text-based review, and has established that using a TF-IDF is an effective way to extract predictive features for this use case.

References

[1] Narendra Gupta, Giuseppe Di Fabbrizio, and Patrick Haffner. Capturing the stars: predicting

ratings for service and product reviews. 2010.

[2] Junyi Wang. Predicting Yelp Star Ratings Based on Text Analysis of User Reviews. 2014.