Will Gleave (solo member)

Final Project Proposal

**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 such as the Washington Post also do not score reviews.

I hope to be able to build a model that will estimate scores for these reviews, based 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 they are given. A website such as Rotten Tomatoes, Metacritic, or any review aggregation service would care directly about being able to put 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 quality 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 allow for better understanding all around between seekers and receivers of feedback.

**2. Define Objectives and Metrics:**

In order to transition to the desired alternative solution, I want to predict the review score for a game review. Since the reviews I plan to use are scored continuously from 0-10, I will attempt to minimize the error between the true review score and the predicted review score.

I plan to use root mean square error as 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.

I plan to use video game reviews from the website IGN.com to build the predictive model. I will need to scrape the website for review text and score data. 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 harder 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 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 RMSE to measure the success of this work. A high RMSE will mean that using TF-IDF for score prediction will be successful, while a low RMSE will mean that it 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 the just using the average score for the dataset. 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 simple word vector. 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 using the average score of the dataset as the predictor.

I attempted using number of different methods to see which methods gave me the best results. In addition to using the word vector as the input to the TF-IDF, I experimented with using n-grams as the input. This typically produced worse performance, with RMSE in the range of 1.1-1.2. I also experimented using topic modeling as a means of dimensionality reduction of the model input. Despite trying different alpha values and topic numbers to tune the LDA, the LDA did not seem to choose topics that differentiated much based on review score. The LDA features did not appear to be 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 using only 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 build 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 | 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 fairly vanilla. From exploration, it does appear that average review scores increased by time, and also differed by game platform. 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. The top median difference between prediction and actual score of .605 is promising. However, the RMSE is higher than the median due to larger errors that increase the average error. Large improvements could be made to the RMSE if the predictions that have the larger errors were more accurate.

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 significantly. The hypothesis may not seem that difficult a bar to clear, but a model with TF-IDF features also significantly outperformed models built using the genre as predictor, or models with topic models as features. There is clearly some predictive value in the TF-IDF representation of data that can predict the numerical score of a review.

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

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.