Amazon Rating Prediction

Peter Brydon
A10463277
University of California, San Diego
9500 Gilman Dr
La Jolla, California 92093
pjbrydon@ucsd.edu

Kevin Groarke
A11882737
University of California, San Diego
9500 Gilman Dr
La Jolla, California 92093
kgroarke@ucsd.edu

ABSTRACT

Identifying user review ratings based off of sentiment analysis techniques is an important topic in machine learning, computer vision, and data science. In this paper we build a model to predict product ratings based off of rating text using a bag-of-words model. The two models tested utilized unigrams and bigrams. The dataset used is a subset Amazon video game user reviews.

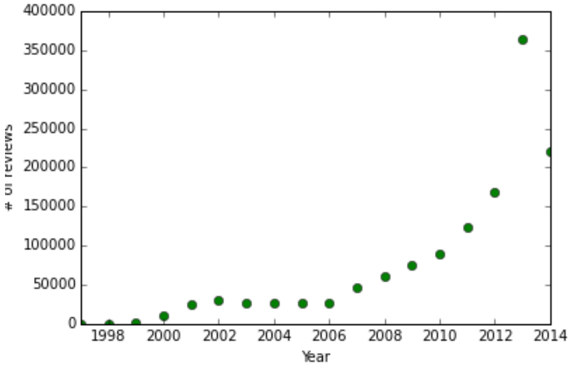
1. THE DATA SET

We used the data set *reviews_Video_Games.json.gz*, which can be found on the link 'jmcauley.ucsd.edu/data/amazon'. The dataset contains 1324759 video game reviews from May 1996 - July 2014. Each review contains reviewerID, ASIN, ReviewerName, helpfulness, reviewText, overall rating, summary, and rating time.

1.1 Basic Statistics

- **Mean Rating:** 3.97875613602
- **Rating Standard Deviation:** 1.3789844278
- **Number of unique items:** 50210
- **Number of unique users:** 826773
- **Top 10 Items - (# of reviews, ASIN, Title):**
  1. (16222, 'B00DJFIMW6', 'Despicable Me: Minion Rush)
  2. (7561, 'B00BGA9WK2', 'PlayStation 4')
  3. (5713, 'B00FAX6XQC', 'DEER HUNTER 2014')
  4. (5490, 'B009KS4XRO', 'BINGO Blitz - FREE Bingo + Slots')
  5. (5190, 'B002VBWIP6', 'Xbox Live 12 Month Gold Membership')
  6. (4638, 'B0055SWM08', 'Quell')
  7. (4510, 'B00CSR2J9I', 'Hill Climb Racing')
  8. (4468, 'B0015AARJI', 'PlayStation 3 Dualshock 3 Wireless Controller (Black)')
  9. (3522, 'B00178630A', 'Diablo III - PC/Mac')
  10. (3290, 'B000FKBCX4', 'Spore - PC/Mac')
- **Top 10 Users - (# of reviews, reviewerID, reveiwer name):**
  1. (880, 'A3V6Z4RCDGRC44', Lisa Shea)
  2. (817, 'A3W4D8XOGLWUN5', Michael Kerner)
  3. (797, 'AJKWF4W7QD4NS', N. Durham)
  4. (521, 'A2QHS1ZCIQOL7E', Richard Baker)
  5. (474, 'A2TCG2HV1VJP6V', Ryan Sil.)
  6. (429, 'A29BQ6B90Y1R5F', Wander)
  7. (338, 'AFV2584U13XP3', Rich!!))
  8. (320, 'A20DZX38KRBIT8', Deimos)
  9. (267, 'A74TA8X5YQ7NE', NeuroSplicer)
  10. (263, 'A2582KMXLK2P06', Bryan)

1.2 Time and the number of reviews



,

Figure 1: Number of reviews on a given year

From Figure 1, we can see that each year, the number of reviews increases linearly, except for 2013 where there is about triple the amount of reviews. 2013 seems to be an outlier, as we can see a linear trend with all the other years and reviews.

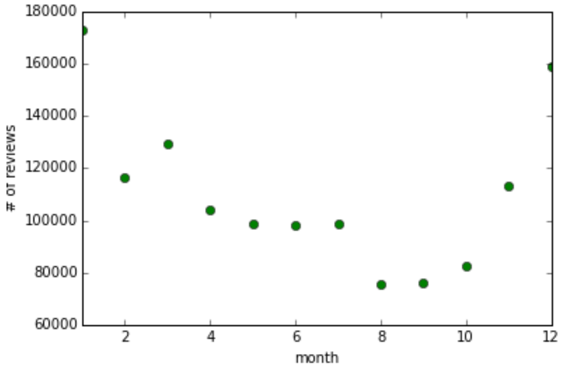


Figure 2: Number of reviews on a given month

From Figure 2, we can see that August, September, and October correspond to the least amount of reviews for those products. This could be due to release date of games or people buying less in those months. The largest amount of reviews come from December through February (end of the year to the beginning). Similarly, this increase in reviews could be associated with the due to the release date of games (especially popular ones) or the buying patterns of gamers.

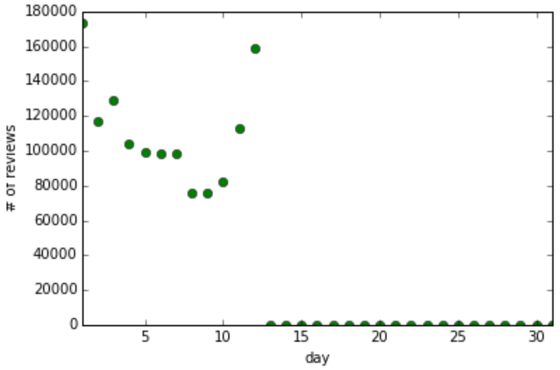


Figure 3: Number of reviews on a given day

Figure 3 shows that the vast majority of games are reviewed within the first twelve days of any given month.

1.3 Time and Average Review Rating

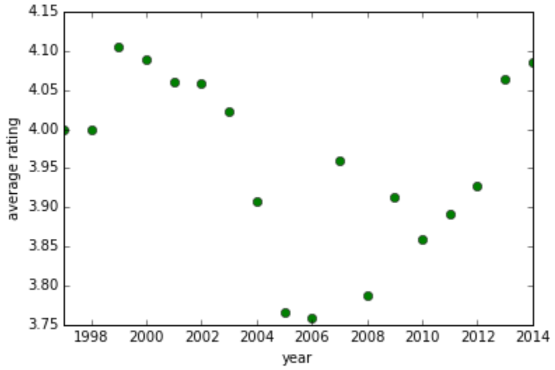


Figure 4: Average rating on a given year

From Figure 4, we can see that between 1998-2002 and 2013-2014, there is a slightly higher rating given compared to the years 2003-2012

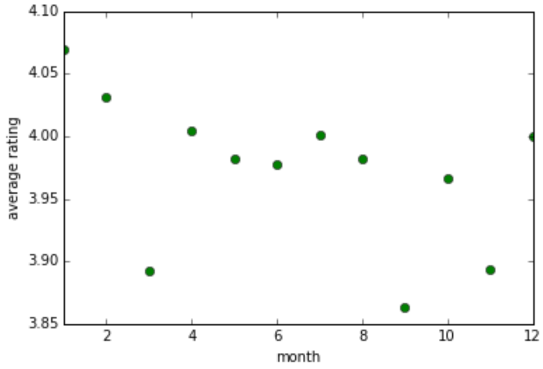


Figure 5: Average rating on a given month

From Figure 5, we can see the ratings given on each month seem a bit sporadic, however there doesn't seem to be any trend on any given set of months.

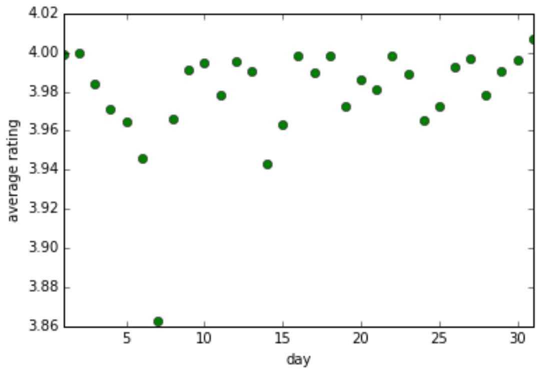


Figure 6: Average rating on a given day

From Figure 6 we can see that the rating given on a particular day of the month doesn't change by any significant amount.

Overall, even though there are some differences in buying patterns between years/months/days, the standard deviation between the average rating is minimal.

1.4 Review Text

The review text analysis is computationally and memory intensive. There is a need to find unigrams and bigrams that tend to explicitly describe positive or negative feelings and the number of unique bigrams tends to dwarf the number of unigrams. Getting a good handle on selecting feature vectors from these sets can therefore be imperative.

1.5 Text Statistics

In the entire dataset there are 151,547,036 words, 1,340,795 unique words, and 12,829,693 unique bigrams. The dataset is split into training, validation, and test sets of sizes 441,565 , 441,501 , and 441,693. The training set has 613,146 unique unigrams and 5,883,988 unique bigrams.

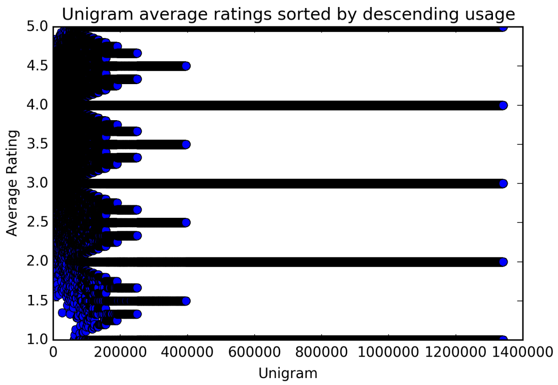


Figure 7: Average rating of unigrams in descending use

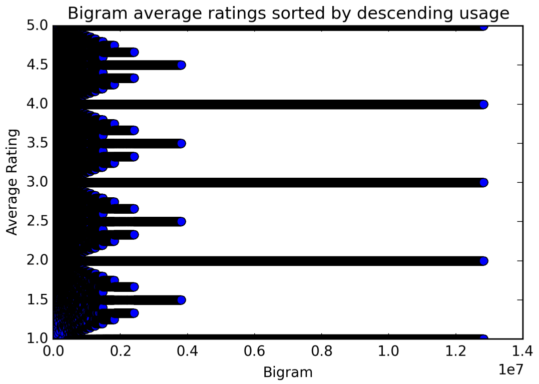


Figure 8: Average rating of bigrams in descending use

From figures 7 and 8, it seems clear that bigrams and unigrams behave similarly from raw average ratings. Among the top used terms, unigrams seem to have more space filled in for ratings above one star, whereas bigrams have a full spectrum.

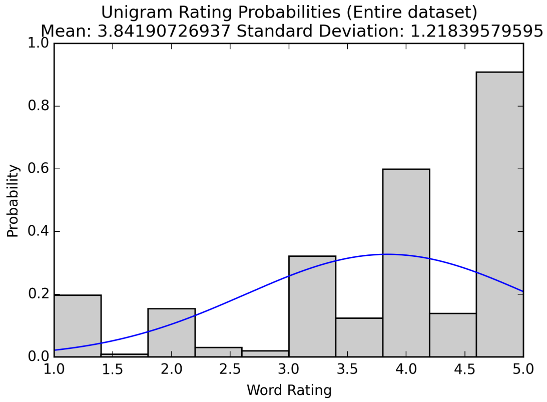


Figure 9: Probability curve of average unigram ratings

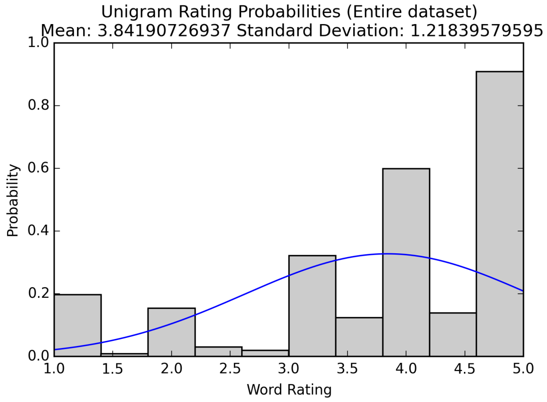


Figure 10: Probability curve of average bigram ratings

Figures 9 and 10 illustrated that the average rating for a term in both sets is roughly around 3.84 in both categories with very similar results.

Unigram and bigram usage throughout documents varies greatly.

Set	μ	σ
Unigram	69.261	3817.644
Unigram first 99 percent	3.77	13.84
Bigram	10.981	468.405
Bigram first 99 percent	2.93	7.28

Figure 11: Unigram and Bigram Document Usage Statistics

The statistics in figure 11 indicate that using the first ninety nine percent of terms in increasing document usage will be a poor predictor. The number of documents using a term indicates how many 'hits' a predictor will have when generating vectors. If documents have no hits, the predictor is only as good as its intersection or other features. Therefore terms in the top one percent by document usage are most helpful.

2. THE PREDICTIVE TASK

The predictive task in this report is to predict the rating of an amazon video game product based on user review data. The user reviews contain information, such as reviewerID, ASIN, reviewerName, helpfulness, reviewText, overall rating, summary, and rating time.

Algorithms that best fit this supervised prediction task of real values includes linear regression and ridge regression. Another important technique that will fit this supervised prediction task is the bag-of-words technique in the form of either unigrams or bigrams.

2.1 Basic Baselines

2.1.1 Constant Feature

Using ridge regression with a constant feature vector of [1] the MSE is 1.90532384435.

2.1.2 Average Rating

Using ridge regression with a constant feature vector of [3.97875613602] the MSE is 1.90532476391

2.1.3 Helpfulness Rating

Using ridge regression with a feature vector of [1, <helpfulness percentage>] the MSE is 1.89281130369

2.2 Time-based Baselines

Time-based baselines were computed using ridge regression with feature vectors consisting of a constant and a binary time based feature. For example, suppose we had a game sold in January, the feature vector would be
[1,0,0,0,0,0,0,0,0,0,0,0,1].

Time	MSE
Year	1.69042711769
Month	1.6997
Day	1.69932740389

Figure 12: Time-based baselines using binary feature vectors

3. THE MODEL

Word usage models were computed using scikit learn's linear model. Specifically, the ridge model was implemented. The intercept parameter was set to false as whenever it was fit to the results it tended to be wore in these models, and the first hyperparameter was left at 1.0. The top 1,000 used unigrams by total times used were selected for the unigram model and the top 1,000 used bigrams were used for the bigram model. Both models were trained using a binary vector.

3.1 Unigrams

Results from the unigram model yielded a mean squared error of 1.144 and 1.145 on validation and test sets. Rounding only served to increase this, as it would increase to around 1.46 in both cases. A set of the top ten best and worst unigrams based on the training data was collected as well. Some of the results were relatively amusing such as the fact ”ea” was treated as worse to see in a review than the word ”return” was.

Top 10 Positive and Negative Baseline Unigrams - (Coefficient, Unigram):

1. (0.36109010871683861, 'love')
2. (0.35835226341835913, 'perfectly')
3. (0.35463839309947515, 'awesome')
4. (0.34360312627929745, 'amazing')
5. (0.33427448325918424, 'great')
6. (0.32993014076918881, 'best')
7. (0.31654419277434226, 'loves')
8. (0.30578075461263243, 'excellent')
9. (0.30378217094241472, 'perfect')
10. (0.29465733392660742, 'addictive')
11. (-0.3999565548537708, 'disappointed')
12. (-0.40783783472372731, 'not')
13. (-0.48470008062982745, 'boring')
14. (-0.49895230473885682, 'poor')
15. (-0.54576150713971505, 'return')
16. (-0.60020502543246623, 'ea')
17. (-0.62582998918787369, 'terrible')
18. (-0.63749205206095394, 'horrible')
19. (-0.79713130834000523, 'worst')
20. (-0.83900036817898282, 'waste')

3.2 Bigrams

Bigrams are thought to potentially hold more sentimental meaning than unigrams, as they can include adjectives or other terms. For instance, the word bad could be used in context of ”very bad” or ”not bad.” A drawback of bigrams is however the fact there are significantly more of them and much fewer usages across documents. An increase from 1,340,795 to 12,829,693 unique terms increases the number of unique terms to sift through by almost a factor of ten.

A bigram regressor was trained on the top 1,000 most popular bigrams found in the training data set. The ridge regressor yielded an mean squared error of 1.360 on the validation set. After naive rounding the mean squared error increased this to 1.678. This was more erroneous than the unigrams approach, and since using that approach yielded similar results in the validation and test sets it’s highly unlikely the test set would have yielded significantly better results than this. Since the graphs of average rating in figures 7 and 8 were quite similar the decision not to run the test set on the bigram case was made in the interest of time. Due to the increased size of the data for bigrams, memory usage would skyrocket when trying to use them to unwieldy levels. As a matter of curiosity however, a list of top ten highest and lowest bigrams was put together in much the same way as they were for the unigrams.

Top 10 Positive and Negative Baseline Bigrams - (Coefficient, Bigram):

1. (0.52067608969945933, (’loves’, ’it’))
2. (0.4910038537857126, (’no’, ’problems’))
3. (0.48652161604763411, (’well’, ’worth’))
4. (0.4759157771489404, (’works’, ’great’))
5. (0.47039406777987114, (’love’, ’it’))
6. (0.46705495562773447, (’is’, ’amazing’))
7. (0.44763304832736517, (’loved’, ’it’))
8. (0.44164147088542005, (’is’, ’awesome’))
9. (0.43691080398465337, (’love’, ’this’))
10. (0.42952777406862186, (’a’, ’must’))
11. (-0.39443296899464225, (’should’, ’have’))
12. (-0.3968185337189703, (’i’, ’tried’))
13. (-0.44776492157009884, (’not’, ’even’))
14. (-0.48600405053816248, (’do’, ’not’))
15. (-0.49949722054289047, (’tried’, ’to’))
16. (-0.50269215355307928, (’my’, ’computer’))
17. (-0.7165061600014021, (’would’, ’not’))
18. (-0.88933530337914468, (’not’, ’work’))
19. (-1.1307812791492855, (’the’, ’worst’))
20. (-1.2814141629791405, (’your’, ’money’))

3.3 Improvement Approaches

Since the bigram model was proving too intense with lower performance than the unigram model, improvements were spent primarily on seeing whether there were any ways to improve the model or fine-tune the results. The fact that the initial statistics were so time consuming to gather meant more preprocessing was necessary if multiple approaches were to be attempted in rapid succession. The size of the files containing review texts needed to be shrunk and words more specifically targeted.

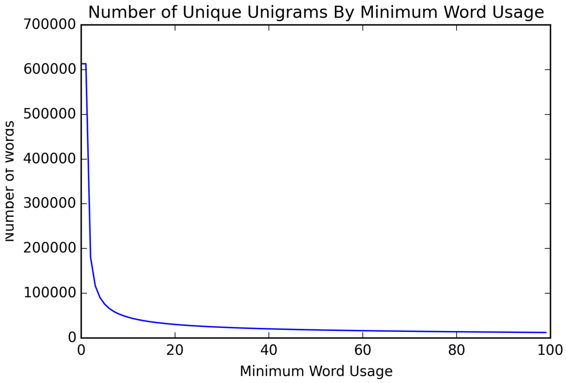


Figure 13: Number of unique unigrams after minimum usage restrictions (Training set)

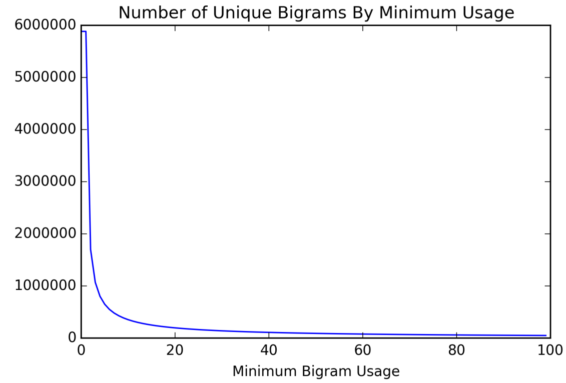


Figure 14: Number of unique bigrams after minimum usage restrictions (Training set)

Figures 12 and 13 show that by merely requiring a word or bigram be used a hundred times in 441,565 reviews the number of unique words and bigrams drops drastically. The curve appears to follow a strong $\left\{n^{-l} \mid l \in \mathbb{R}^{+} \wedge n \in \mathbb{Z}^{+}\right\}$ pattern, dropping off significantly quickly.

A minimum usage parameter based on this was initially used as an attempt to reduce text sizes. The data structures implemented required a significant amount of memory and basically any reduction would help with predictors. The results however, were not particularly thrilling and it only shaved less than 10% off of the file sizes.

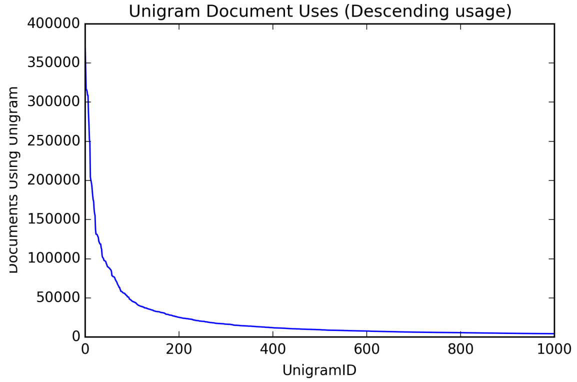


Figure 15: Document Usage of Unigrams (Training set)

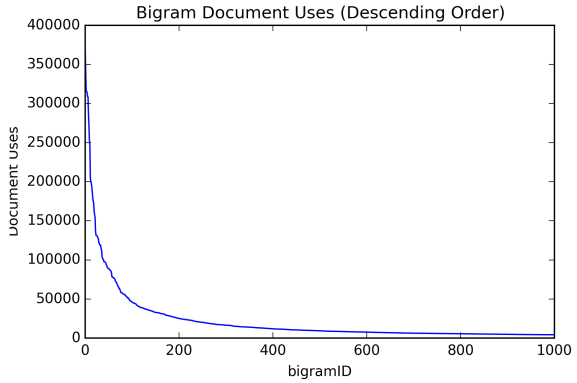


Figure 16: Document Usage of Bigrams (Training set)

Figures 15 and 16 explain the disappointment with the shrinkage attempt. The fact that document usage drops so quickly meant that by eliminating a bunch of words nobody ever really uses the size would of course not drop. It also indicates that models would be ineffective at targetting reviews if the words they are trained on are not used enough. For further compression an encoding system assigning integer numbers to each word was able to garner a further 33% compression. Unfortunately the memory requirements once loaded were still excessive for any processing.

Attempts to improve the words model started with trying to get variation and target words with higher coefficients in the models.

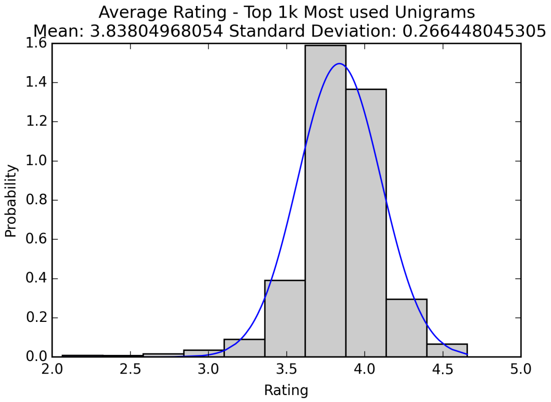


Figure 17: Histogram - Rating of top 1,000 Unigrams (Training set)

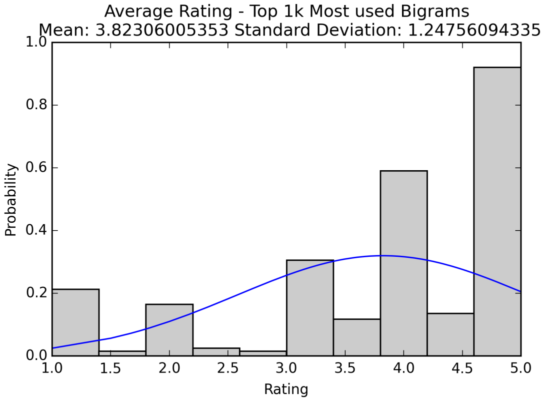


Figure 18: Histogram - Rating of top 1,000 Bigrams (Training set)

Figures 17 and 18 illustrate the data sets and their variation. If more time were alotted it would probably be best to include unigrams for ratings 3-4.5 and bigrams to try and identify lower and higher ratings.

However it was found that due to memory and time constraints it would prove to be impossible to train a significantly more accurate predictor using all 441,565 training reviews.

There were minimal losses in replacing the unigram vector with the top 200 and lowest 200 of the first 1,000 unigrams. However the results were still worse by a factor of around 0.07. Attempts to select models where there were more even distributions of average ratings throughout the most popular terms also yielded minimal improvements. Referencing back to figure 17, there was significantly low variation in the top 1,000 unigrams. Essentially most of the unigrams referred to a range between 3.4 and 4.2. However, as bigrams proved too complex to train on the size of the dataset the potential improvement they could give was not looked into.

The last thing that was done is an attempt to enhance the predictions by seeing if the predictions being made by the unigram predictor could be improved on with simple post-processing.

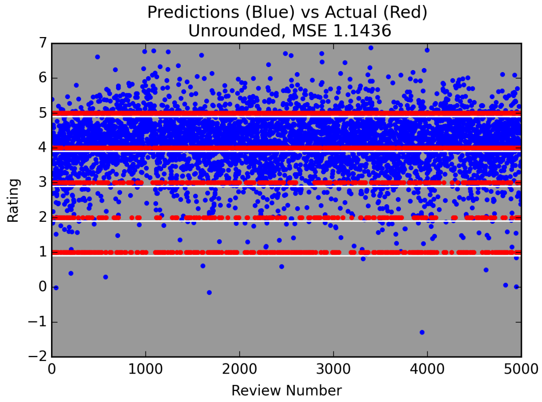


Figure 19: Prediction Analysis (Test set)

Figure 19 indicates a problem that could be solved simply by rounding any prediction below 1.0 to 1 and any prediction above 5.0 to 5.

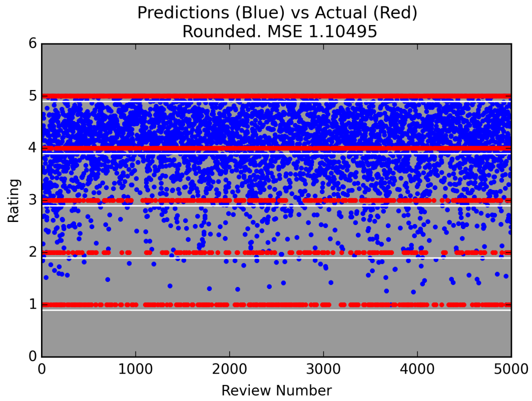


Figure 20: Rounded Prediction Analysis (Test set)

.

As might be expected this yielded an improvement by reducing the mean squared error to 1.112 on the validation set and 1.106 on the test set. A summation analysis of the ratings also seemed to indicate that this model was at first guessing too high most of the time in order to preserve accuracy based on the fact the average review ratings are closer to 4.

$\sum_{i=0}^{n} (\text{predictionsValid}_i - \text{ratingsValid}_i) = 46.38481114$
$\sum_{i=0}^{n} (\text{fixedPredValid}_i - \text{ratingsValid}_i) = -10815.55909824$

After rounding however, the model seems to have a problem with guessing too low on average.

Range	Pr < Rat	Pr > Rat	P(Pr>Rat)
$4.5 < x < 5.0$	67815	17655	≈ 0.79
$4.0 < x < 4.5$	41321	85508	≈ 0.67
$3.5 < x < 4.0$	28343	61950	≈ 0.69
$3.0 < x < 3.5$	29768	20957	≈ 0.41

Figure 21: Validation Prediction Statistics

Figure 21 was created in an attempt to analyze frequencies of rounding errors. Essentially, just how likely is it that our validation set predictions were too low or too high? However, since the probabilities were not one-sided enough in most aspects it would not be particularly wise to simply round. Even the 79% probability of being correct in rounding up from 4.5 did not strongly improve the mean squared error. As a result, it would seem that another model would have to be generated to gain much improvement or the error pattern would have to be identified.

Models such as KNN or SVM were considered and attempted but the runtime of the KNN and SVM algorithms were too high to be ran on the data. The general idea is that they would have an easier time returning a flat label and if trained extremely well may have significantly better accuracy than a linear model. Unfortunately it would seem that the final mean squared errors for the unigram predictor would have to be 1.112 and 1.106.

4. LITERATURE

The dataset used in this article is described in section 1 [1]. The dataset was found at the link 'jmcauley.ucsd.edu/data/amazon'. The article used this data to create image-based recommendations on styles and substitutes.

Another article that used this data set is 'Inferring Networks of Substitutable and Complementary Products' [2]. The amazon product data set was used to recommend complimentary and substitute products based on aspects of the product reviews, such as review text, price, and brand.

Another article that performed the same predictive task on a different portion of the data set (automotive reviews) also employed the Bag-of-Words method as the main method for prediction [3]. However our model differs because it uses ridge regression instead of linear regression. The ridge regression along with rounding seems to improve the overall model with an mse of 1.112, while the linear model had an mse around 1.74.

5. RESULTS

Of the different models tried, the bag-of-words method (utilizing popular unigrams or bigrams) produced the most accurate predictive results.

Models that didn't work well were time-based models. These models utilized the time a user reviewed a product (year, month, day). They did not serve as good predictors because the variance in the average rating between each year, month, or day was relatively small. As a result any model based off of time wouldn't make sense because time (as shown in the data set exploration section) has a relatively low correlation with ratings. Thus our small improvement makes sense, since there is a small correlation with user review ratings. These are most likely associated with the times at which highly rated games tend to be rated.

Although the bag-of-words model works well in general with user review rating prediction, unigrams produce the most accurate result. The reason bigrams didn't produce as accurate of a result is likely because there are fewer usages throughout the documents and as a result they tended to capture a significant amount of variance that may only have been present in a select few. Also, many of the bigrams could be interpreted negatively or positively based on the context. Like most unigrams, bigrams can be used as positive and negative. Some may be more likely to be negative or positive however that can also introduce noise if a certain subset of users has a different way of speaking or likes to prefix bigrams with negations or sentiments that may result in a different point of view. The bigram feature may have been an improvement over the unigram model if there was a bigger review space and we had a super-computer to crunch all of the bigrams for our bag-of-words model. This may have improved the model since bigrams tend to be more descriptive than unigrams.

The unigram results had at least a 15.89% performance increase over bigrams and unigrams were significantly less difficult to process. The bigrams may have higher scores in the regressor, but they do have less usage. If it wasn't for the extremely long runtime of the algorithms to generate them it would be worthwhile to consider a fusion and see what happens but a better approach may be to better target unigrams that would lead to a better predictor or find some additional features.

In conclusion, popular unigrams seemed to be an extremely useful predictor for ratings. Bigrams may have been useful because of their larger variance but were a bit intense to compute for. The end result being a predictor that can guess with a mean squared error of approximately 1.11 or 1.10 on a dataset of approximately 441,000 reviews.

6. REFERENCES

[1] J. Shi A. van den Hengel McAuley, C. Targett. Image-based recommendations on styles and substitutes. *SIGIR*, 2015.

[2] J. Leskovec J. McAuley, R. Pandey. Inferring networks of substitutable and complementary products. *Knowledge Discovery and Data Mining*, 2015.

[3] Richard Park. Assignment 1: Predicting amazon review ratings. 2015.