AM 221 Milestone 2

Arjun Mody & Michael Zochowski

April 2, 2014

1. **Data sets**. We collected several datasets for this project. The one we use for this project is a collection of Amazon reviews regarding software products. The data set can be found through SNAP (Stanford Network Analysis Project) at the following link (https://snap.stanford.edu/data/web-Amazon.html) under "Software Reviews." We may extend our analysis to other product types if we have the time to compare predictors (through SNAP we have access to all of the product types listed). We also obtained data from the Yelp Dataset Challenge (http://www.yelp.com/dataset_challenge), but we determined that the Amazon data has metadata of higher quality.

   Each item in our data set contains the following fields:

Field	Example
product/productId	B00006HAXW
review/userId	A1RSDE90N6RSZF
review/profileName	Michael C. Zochowski
review/helpfulness	121/221
review/score	5.0
review/time	1042502400
review/summary	Enjoyable
review/text	My five-year-old had a great time playing this computer game.

   Rating is an integer between 1.0 and 5.0. Review helpfulness denotes that X out of Y found the review helpful. Review summary is the title that appears above the review text when users are browsing through product reviews.

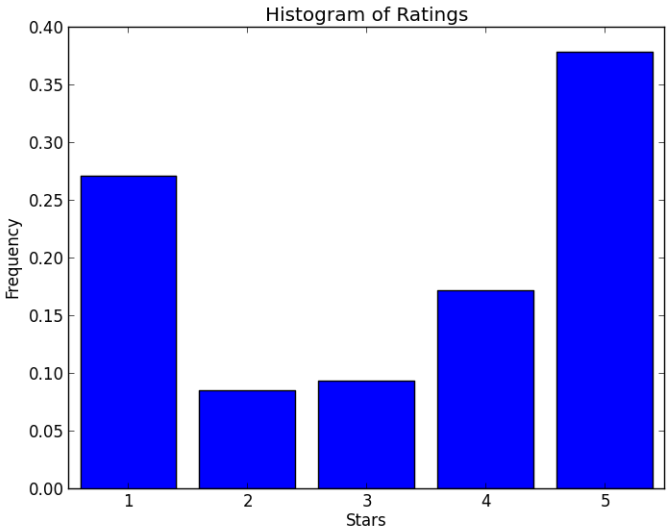
2. **Basic Analysis**. We compute some simple summary statistics on our data set, given in the following table:

Number of Reviews	95,084
Avg. Rating	3.3008
Avg. Helpfulness	9.65 / 11.98
Avg. Helpfulness %	0.769
Avg. No. Words / Review	127.1

Note that the average helpfulness percentage is slightly lower than the average numerator value (9.65) divided by the average denominator value (11.98) since we count reviews as 0/1 if zero people out of zero find the review helpful.

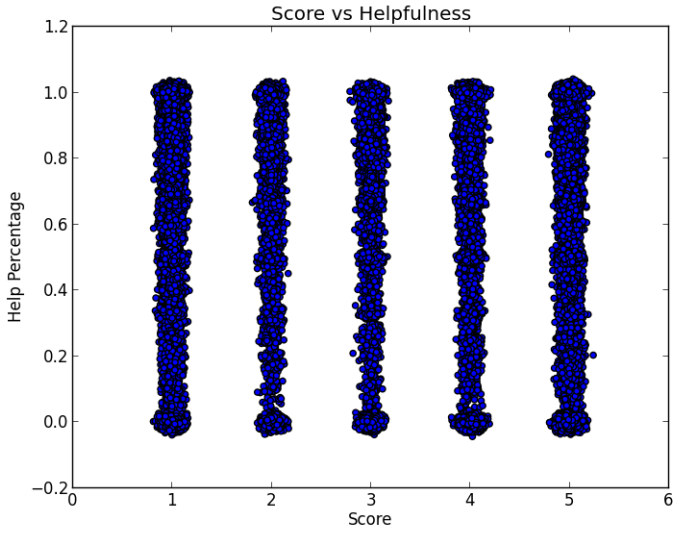
Below is the distribution of ratings:

Rating	Frequency
⋆	0.2713
⋆⋆	0.8493
⋆⋆⋆	0.0935
⋆⋆⋆⋆	0.1721
⋆⋆⋆⋆⋆	0.3781

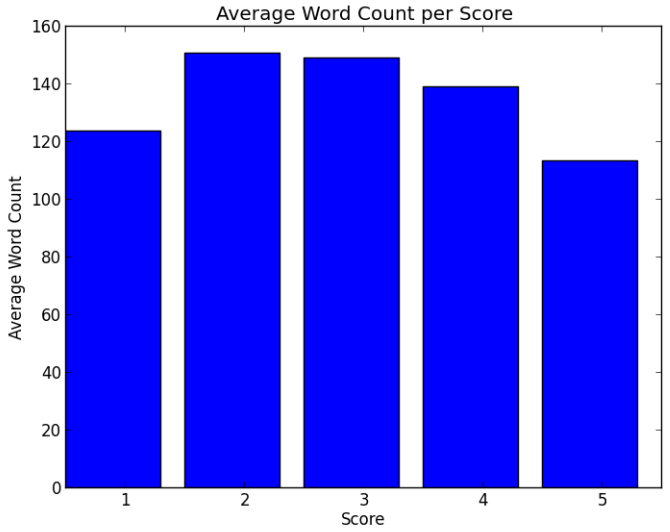


We were also interested in whether there is any correlation between how helpful a review is and the score of the corresponding reviews. Using a method of plotting known as jitter (whereby small deviations are introduced as to make it easy to see clusters of points when many of them will have the same value), we produce the following graph. It tells us that there is no meaningful correlation between how helpful people found a review and how many stars the review had (people seem to find all types of reviews equally helpful). This contradicted our intuition that five-star

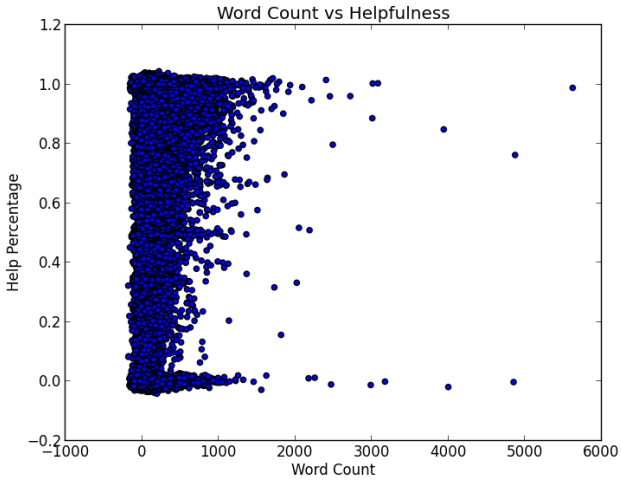
and one-star reviews would be considered "more helpful" since they would provide strong reasons either for or against buying a product.



We can also ask whether there is a correlation between the length of a review and its rating. We find that one and five-star reviews have significantly fewer words than two to four-star reviews.



Finally, we can ask whether there is any correlation between the length of a review and its helpfulness. We find a small correlation of 0.2163, and the scatter plot is shown below:



3. **Benchmark**. We will include this in our next milestone. We've begun to explore using the library at this link to find a benchmark for multi-class classification.

4. **Model**. The optimization problem we want to solve builds on the SVM model we used for Homework 2. We will be extending the SVM classification approach (formulated as a linear program) that we developed in the class to multi-class classification. One common way to classify a string into one of K classes is the "one-against-all" approach, which involves K SVMs where each SVM X_k determines whether the input is more likely to be in class k or in any of the other classes. Then, the SVM for which the input has the highest score (i.e. is the farthest distance above the supporting hyperplane) will determine the input's class. There are other ways to perform the classification as well, some of which use $K(K-1)/2$ SVMs (see paper here). Our features will also consist of unigrams, bigrams, and trigrams, and we hope to measure how performance improves (or does not) as longer, more complex phrases are used as features.

   Other approaches we will consider are the "one-against-one" method, which compares each pair of categories, the directed acyclic graph method, and the error corrected output coding.

   Moreover, we will be using some basic sentiment analysis to pre-process the reviews, determining which sentences are "objective" and which are

"subjective," using only subjective statements in determining our feature vectors.

5. **Third Milestone**. For the next milestone, we would like to have a basic "one-vs-all" classifier working (which will involve constructing 5 SVMs) based on unigrams, and compare its results to a benchmark (some out-of-the-box multi-class classifier that we can find).