|  |
| --- |
|  |
| **Usefulness of Yelp Reviews: An Extrapolation of Leatherback Turtle Breeding Patterns** |
|  |

**Alan RidwanSuman Jandhyala**

Department of Computer Science Department of Computer Science

University of Washington University of Washington

*marksein@cs.uw.edu* *sumanvyj@cs.uw.edu*

**Abstract**

This paper tests the ability to use machine-learning algorithms to predict the usefulness of Yelp consumer reviews. The primary algorithm used in this experiment is stochastic gradient descent for logistic regression and Least Square Error to measure results. The raw review data was provided by Yelp through Kaggle for a Machine Learning contest that has already ended [1, 2]. The paper will focus on feature selection, logistic regression, result discussion, and future recommendations.

**1 Intro**

Many companies have profited from the increase of available information on the Internet. Recommendations and reviews which people would once in the past only take from friends now are readily available on websites like Yelp. While having these reviews is good, a huge problem in the market lies in finding the reviews that are actually useful. The natural process of surfacing useful reviews takes time, as more and more users read and vote on each one. Unfortunately the market is very competitive and users are not very patient; it would be a great advantage for Yelp to know which reviews are more useful than others even before any users have to read and vote on anything [1].

In order to predict usefulness, it is essential to predict what is likely to make a review useful by feature selection. Feature selection allows us to select a subset of information that we think is relevant to the problem we are trying to answer. In this case all features selected are related to the usefulness of reviews. Using our selected features as vectors, we will train a logistic regression model using stochastic gradient descent and see what kind of prediction error results.

This paper is organized in four sections. Section 1 is the Introduction. Section 2 focuses on the data used and the features selected. Section 3 is about the experiment and results. Section 4 concludes the paper and recommends a future course of action.

**2 Data and Features**

The data that Yelp provided on Kaggle contains information on 11,537 businesses, 8282

checkin sets, 43,873 users and 229,907 reviews. Each of these 4 data types is stored in a separate JSON-formatted file. Upon getting the data we immediately parsed and stored each file into a separate but related SQLite database. This allows specific data to be easily queried for during feature selection. We mainly focus on three data types: “business”, “review”, and “user” (Yelp-provided schemas are shown below; irrelevant fields are omitted) [2].

**business**

'business\_id': (encrypted business id)

'name': (business name)

'stars': (star rating, rounded to half-stars)

'review\_count': review count

**review**

'business\_id': (encrypted business id)

'user\_id': (encrypted user id)

'stars': (star rating, rounded to half-stars)

**review (continued)**

'text': (review text)

'votes': {(vote type): (count)}}

**user**

'user\_id': (encrypted user id)

'name': (first name)

'review\_count': (review count)

'average\_stars': (floating point average, like 4.31)

'votes': {(vote type): (count)}

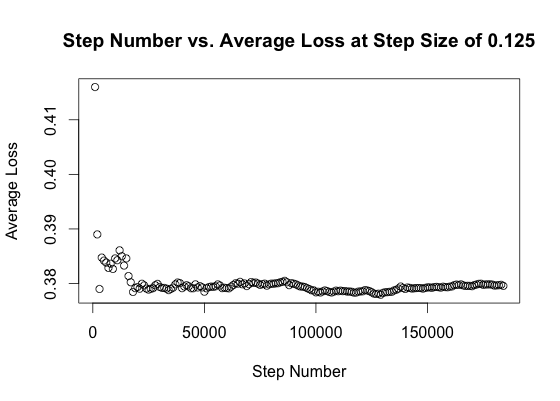
Vote types are ‘funny’, ‘useful’, and ‘cool’.

There are a number of different features that we have hypothesized to have an effect on how useful a review can be. First, we assume the length of a review would affect how useful a review is, because there would be more information in longer reviews. Second, we assume the rating in a review would affect the usefulness of a review, because the more opinionated a review is the more information it might give. Next, the fraction of useful reviews written by a user would show how likely it is that the user would write another useful review. Then, the number of reviews a business has would also affect how useful the review is. Aside from the above features, we also include Flesch–Kincaid reading level and reading ease as factors for usefulness because readability and understandability seem likely to affect how useful someone would find a review.

**3 Experiment & Results**

In this experiment, we determined that if a review had at least 2 votes of usefulness then the review would be considered as useful; 2 votes means that at least 3 people agree the review has good information (including the reviewer), which is reasonable for a minimum threshold.

The figure below shows the average loss during training plotted against the step number (increments of 1000). The average loss converges to ~0.38 quickly relative to the total number of steps performed.



Out of the step sizes 0.5, 0.25, 0.125, and 0.1, the step size 0.125 had the lowest sum-squared error of 13888 (versus 17105, 15565, and 17428 for step sizes 0.5, 0.25, and 0.1, respectively).

With a gradient descent step size of 0.125, we obtained an L2 norm of 2088.29 and a sum-squared error of 13888. The vector of weights for features is shown below.

**Base**: -56.28003

**Length**: -81.06665

**Reading ease**: -336.45032

**Reading level**: -1273.93582

**Rating of review**: -305.03348

**Ratio of useful reviews written**: 1586.91766

**Number of reviews for business**: -61.15562

Based on the weights, we found that reading ease and a user’s ratio of useful reviews written greatly affect the usefulness of a review. When a review is easier to read it is more likely to be useful, and users who write useful reviews are more likely to continue to write useful reviews (no surprise here).

The SSE of 13888 indicates that 13888 mistakes were made out of a total 45908 testing data points. This amounts to an error rate of ~30.3%.

**4 Conclusion:**

Initial results are promising in the sense that there are indications this is a tractable problem. Even with a very simple logistic regression implementation and not very many features, the error rate was lowered to ~30.3% from the 50% you might expect from a purely random model. However, it’s clear that there is a lot of room for improvement.

There are several different courses of action to pursue going forward to reduce the error rate. For one, the feature set could be expanded to include features like review sentiment, keyword presence, and word/phrase repetition. Another course of action would be to make use of another model (like perceptron/SVM) and potentially use boosting.

Beyond simply improving the model’s efficacy, it would be interesting to examine not only how useful a review is, but also how the different classifications (funny, useful, and cool) interact with each other. To this end, we are considering the use of a multiclass SVM, or even a structural SVM to support multiple labels. It would also be instructive to ignore Yelp’s provided labels and use an unsupervised learning algorithm to see what types of labels can be discerned from raw data.

**5 References:**

[1] Kaggle/Yelp. 2013. Yelp Recruiting Competition. (March 2013). Retrieved May 24, 2013 from <http://www.kaggle.com/c/yelp-recruiting>.

[2] Yelp. 2013. Yelp Dataset Challenge. (March 2013). Retrieved May 24, 2013 from <http://www.yelp.com/dataset_challenge>.