EECS 349: Yelp User Usefulness Prediction Model

Joseph Lee, Hyung-Soon Daegon Kim, Kevin Ye, Ji-hoon Kim
hyungkim1.2013@u.northwestern.edu
Northwestern University

Extended Abstract

The goal of this project was to determine if a user was useful or not as a reviewer on Yelp. The more useful a user is, the more "useful" votes he or she would have received from other users. This is important since businesses sometimes create fake accounts to boost their reputations online and increase their ratings or decrease the rating of their competitors.

We used attributes and data from the Yelp Dataset Challenge including attributes such as user account age, total number of reviews per user, and number of friends. Natural language processing was used to generate further attributes on the data: average review length, and average swears per review being among a few. The machine learning algorithms explored were nearest neighbor, decision tree, naive Bayes, and Bayes net. Both training and validation sets were designed to consist of equal numbers of "useful" and "unuseful" user classifications. This created a dataset of 19000 users when "useful" was defined as a user having at least 1 useful vote per review.

J48 proved to be the most effective with an accuracy of 75.07% using 10-fold cross validation and 71% on the validation set. Utilisation of attributes that are more tightly correlated with our target attribute such as "cool" votes or "funny" votes returns much higher accuracies, but we have concluded these to be misleading and realistically not very useful.

EECS 349: Yelp User Usefulness Prediction Model

Detailed Analysis Report

Machine Learning Software Packages

Software packages that the team has used include the following:

- Anaconda Package - A collection of various science packages for python. From the Anaconda Package we have used numpy, sklearn, pandas, textblob, NLTK.
- Python 2.7 / Python 3.4
- Weka

Problem Outline

Our task is to create a predictor of the helpfulness of Yelp users based on a user's past Yelp reviews and account history. **This task is important because** many reviewers on yelp are false accounts made to boost a business' review score - this would help to filter out such accounts. In addition, Yelp could use the information to display the posts from the most helpful users near the top of reviews for a business or to create a "trusted reviews" section.

Dataset: Attributes / Features

We have divided our problem into 2 main approaches: inclusion of community provided attributes (votes from other users) and exclusion of such attributes. Exclusion allows investigating into a portion of the data untouched by the community - human provided information comes at an expense and can be difficult to obtain; being able to extract information without such need would prove to be of meaningful use.We deemed a concurrent investigation and comparison of the 2 would be interesting to pursue.

[**Input**]

The full list of attributes used in the exclusion set are:

- Review count of user
- Yelping for period (how long has it been since user registration)
- Stars given by user (average and std)
- Polarity of texts of user (average and std)
- Subjectivity of texts of user (average and std)
- Review text word count (average and std)
- The star rating of the business the user reviewed (average and std)
- The number of reviews of the business the user reviewed (average and std)
- Swear count (average)
- Vocabulary density (average: ratio of unique words to the total number of words in a review)
- Punctuation density (average: ratio of punctuation marks to total words in a review)

The inclusive set adds 3 additional attributes to the ones above:

- Number of "funny" votes for reviews (average)
- Number of "cool" votes for reviews (average)
- Number of fans

[**Output**]

- "User usefulness"
- : votes.useful / review count (boolean classification against a threshold)

Defining usefulness of a user was the key aim to our task. We settled upon a normalised ratio between the number of 'useful' votes a user gained from their reviews against their total number of reviews. We then decided on a threshold to create a boolean 'user_helpful' value that we could use for binary classification.

However, we were given the impression that the threshold may skew our results in a certain direction. Hence we went the direction of producing multiple results of the same process, just using different thresholds to divide the data.

Training / Validation partition

The team obtained data sets from the annual Yelp dataset challenge. The original user dataset consists of roughly 39,000 data points, the review dataset of over 1,500,000, and the business set of roughly 60,000. The user dataset was joined with the reviews and enterprise data sets to form a larger user dataset of one file. Natural Language Processing and insights made into attribute relationships throughout our iterations were used to produce additional custom attributes in an attempt to extract more information from the data collected.

However, we realised during our iterations that having a significantly low partition of 'true' or 'false' data points could skew our models. Hence, we went through further processing to make sure that we had the same number of 'true' and 'false' data points for each threshold test.

The overall training and validation set was then made by splitting the resulting threshold set into two equal parts. We felt having a roughly equal data set would give us statistically more stable results. Our resulting dataset size for threshold 1 was 19,000 (for both training and validation) and roughly 7000 for threshold 2.

Algorithms For Training

Given our data set, we created J48 decision trees, naive bayes, bayesian net, and k-nearest neighbor (IBk) learning models using WEKA and Sklearn. The baseline was established using WEKA's ZeroR.

<u>Decision Tree Classifier (J48)</u>

J48 was selected to see if there existed attributes of dominant information gain. This model utilizes an ID3 heuristic that uses the entropy and information gain of the data set as the primary splitting requirements for the decision tree classifier.

<u>Bayes Net & Naive Bayes</u>

Probabilistic models have become increasingly popular due to their ability to capture non-deterministic relationships among attributes extracted from real world domains. A bayesian network generally uses graphical elements to efficiently represent a joint probability distribution over a set of attributes, whereas naive bayes classifiers uses Bayes Theorem with the assumption of independence between features. For our project, Bayes

Net was chosen to see if there existed causality relationships between attributes in conjunction with Naive Bayes for comparison.

K-Nearest Neighbours Classifier (IBk)

K-nearest neighbours classifier are able to select appropriate value of K based on cross-validation. In short, it is a non-parametric method with applications for classification and regression problems that is often useful for pattern recognition applications. For this project, IBk was selected to see if there exists any general correlations in data based on feature space and distance.

Validation Results

As expected, the results from the "inclusive" set had higher accuracy (approx. 95% for each threshold) - the funny vote and cool votes indeed seemed tightly coupled with our target attribute. Decision trees (75%) and Bayes Net (68%) performed the best out of all the applied machine learning algorithms overall. J48 has a slight loss of about 4~8% on the validation set depending on the threshold, but seems to show that it was less of an overfit, which would be important for larger data sets.

Taking a closer look at the nodes the j48 trees would split on, interestingly, it seems that the number of stars of the business reviewed, as well as the number of the reviews the business had, was a factor on deciding user usefulness. The data seems to express that under similar user profile conditions, users were more likely to say that a review was helpful if the business being reviewed had a higher number of stars or if the business already had a large number of reviews.

The attribute which provided the highest information gain for decision tree algorithms using the inclusion set were related to other community-given votes. This supports the idea that useful votes is correlated with funny and cool votes. The decision tree results show very high accuracy for inclusion sets, but these results are lower in meaning due to the relationship between votes and our output attribute, which is based on one type of these votes.

For the exclusion set, the decision tree algorithms found highest information gain from age of user accounts and total reviews published. Users with comprehensive Yelp histories are deemed more trustworthy than "one-and-done" users.

In general the accuracies for threshold 2 were about 10% greater than in threshold 1 in the training sets. In the validation sets, the accuracies were about the same - thus suggesting the existence of a correlation regardless of threshold bounds. In terms of the validation set, j48 seemed to outdo BayesNet for the inclusion set (95% vs. 68%) while both the models rested closely against the 70% line for the exclusion set. The findings conclude a useful or significant relationship between the user provided votes and our desired output function, whereas the exclusion of user provided votes results in no significant attributes nor relationship. However, a correlation may be easily determined by a combination of attributes (ex. review count and reviewed business information) and latching upon their relative correlations.

Suggestions for future work

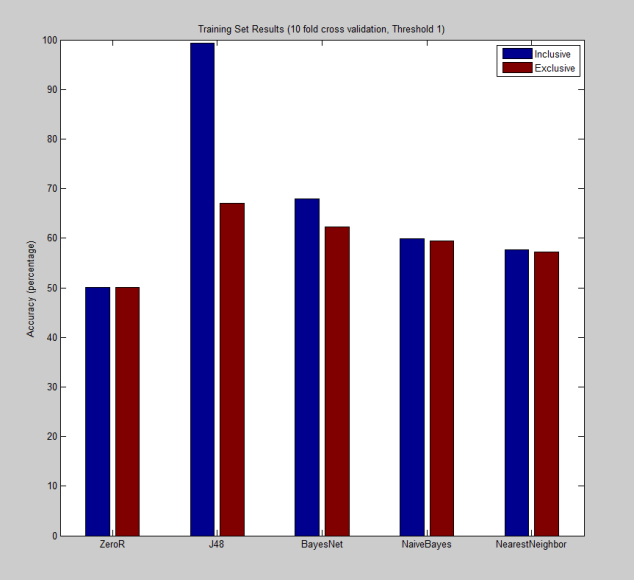
From observation, Yelp seems to employ a custom algorithm to differentiate between good recommendations and bad. A comparison into this algorithm may prove to be of use and may inspire the generation of further possible attributes.

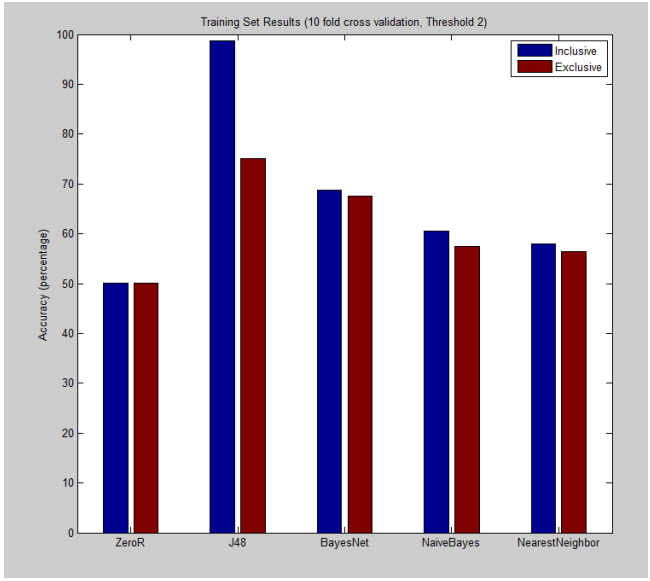
Natural Language Processing did not yield sufficient results for our project. If given an extended time frame, it might be fruitful to scrape the websites of businesses and compare that with text information in the reviews. This approach may provide a useful dataset to construct an insight into 'market speech' - a classification we might be able to apply to the text data to filter out more accounts.

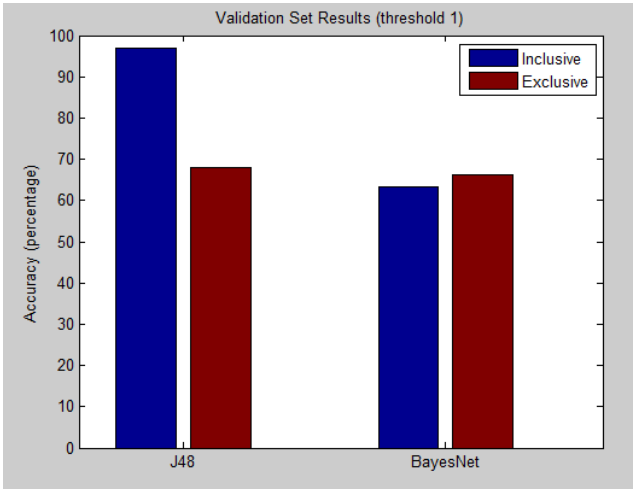
Finally, judging whether a user is helpful and judging whether a review is helpful are 2 facets of the 'usefulness' question that our group did not pursue to compare - it would be interesting if further work could be done in this direction.

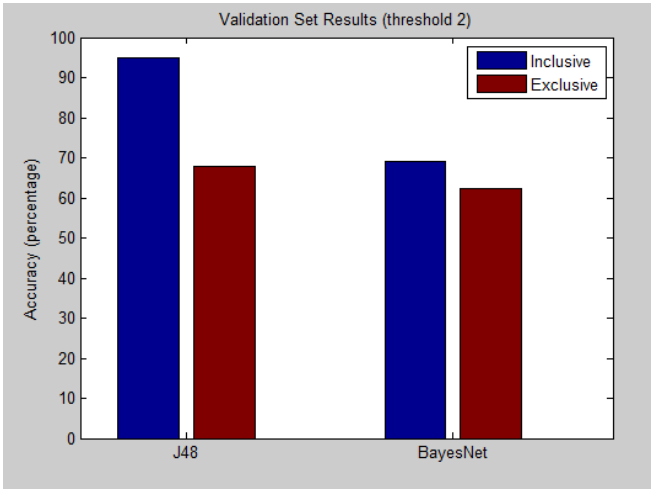
Appendix A: Iteration Graphs

Threshold 1 requires a user to have an average of 1 useful vote per review to be deemed “useful.”
Threshold 2 requires a user to have an average of 2 useful votes per review to be deemed “useful.”
“Inclusive” means that the algorithm includes attributes from other users such such as “cool” votes.
“Exclusive” means that the algorithm only uses attributes that do not depend on other users.









Appendix B: Iteration Log

The below is a snippet from some of our latter iterations

<**Threshold 2**> (all quotients at least or over 2 are deemed 'True', others 'False')

Classifier	**10-fold Cross validation accuracy (Inclusion)**	**10-fold Cross validation accuracy (Exclusion)**
rules.ZeroR	50.0077%	50.0077%
trees.J48	**98.6751%**	**75.0732%**
rules.BayesNet	68.8338%	67.6013%
rules.NaiveBayes	60.5454%	57.5258%
lazy.IBk	57.8801%	56.4474%

J48 node order:
Inclusion: votes.funny > review_count > yelping_for_period
Exclusion: yelping_for_period > review_count > review_stars_avg > biz_stars_avg

[Validation Set Results] - application of training model on validation set)
: ***Inclusion Results***

trees.J48 **94.839 %**
rules.BayesNet 67.8169 %

: ***Exclusion Results***

trees.J48 **71.0032 %**
rules.BayesNet **64.3891 %**

<**Threshold 1**>

Classifier	**10-fold Cross validation accuracy (Inclusion)**	**10-fold Cross validation accuracy (Exclusion)**
rules.ZeroR	50.0026%	50.0026%
trees.J48	**99.258%**	**67.0904%**
rules.BayesNet	67.9051%	62.2594%
rules.NaiveBayes	59.8879%	59.5143%
lazy.IBk	57.6981%	57.2051%

Note) training / test dataset size: 19,272

J48 node order:
Inclusion: votes_total > review_count > biz_stars_avg
Exclusion: yelping_for_period> review_count > biz_review_count_avg

[Validation Set Results]
: Inclusion

trees.J48 **96.8813 %**
rules.BayesNet **67.9248 %**

: Exclusion

trees.J48 **63.5411 %**
rules.BayesNet **66.2302 %**