Boosted Decision Tree Regression to Predict Yelp Review Stars from Review Text

Apurva Pathak
A53097569
UC San Diego
appathak@ucsd.edu

Chaitanya Baratam
A53104872
UC San Diego
cbaratam@ucsd.edu

Shrestha Malik
A53087000
UC San Diego
shm039@ucsd.edu

Phani Teja Marupaka
A53087859
UC San Diego
pmarupak@ucsd.edu

ABSTRACT

In this project, we consider the problem of predicting restaurant ratings solely from the text reviews given by the user. This is a challenging text mining problem and it has many applications on online review platforms. The project involves evaluating different feature extraction methods: Bag of words and Vowpal Wabbit. These extracted features are trained using machine learning techniques such as Linear regression and Boosted Decision Tree Regression for rating prediction. The motivation is to find the combination of best machine learning technique and feature extraction method to solve the problem. In addition, we analyzed how the sentiments of people vary with geographical location and time. Further, we used the review text to group users based on their tastes and preferences.

Keywords

Sentiment analysis; regression; text mining; boosted decision tree regression; vowpal wabbit

1. INTRODUCTION

Now a days, many people use online websites to select the restaurant that they want to visit. As they don't have the time to read the reviews written by the users, they depend mostly on the rating given by the user. But the problem with the ratings given by the users is that, each user will be having different standards for rating. For one user, the restaurant needs to be excellent to give a 5 star rating whereas another user might give 5 star rating to a restaurant which is just good. Yelp, most popular business ratings and reviews website, has large amount of interesting data in the restaurant domain. Yelp dataset has the problem of user bias and we wanted to explore this dataset to see how we can predict the rating based on review text. Figure 1 clearly shows the bias of the two users. Both the users praised the restaurants with similar words and both of them found the restaurant to be very good. But the ratings that they gave to the restaurant are very different. Our goal is to lessen the effect of these biased reviews by training our model on large data set. So, we decided to use only the review text and not the user or item features for prediction. This is also aligned with one of the challenges of Yelp dataset 2016. In addition to this, we wanted to check if people in different locations write reviews differently. Further, we analyzed ratings from different times to see if there is a change in the way people wrote reviews across different years. We also explored clustering of users based on their taste and preferences. The assumption is that users with similar tastes will rate a given restaurant in a similar way which will be also reflected in their review text. Such users should be part of the same cluster.

Figure 1: Screenshot of two similar Yelp reviews with different ratings. Picture taken from [3].

2. LITERATURE REVIEW

Collaborative Filtering, Matrix Factorization, Latent Factor Modeling and Hidden Topics [7] are some established and well researched tools for predicting rating and recommendation and use hidden user and item features. However our focus was on predicting a context aware unbiased rating for the item based only on the review text. Similar work has been done for predicting helpfulness of the review [4][10]. We referred many state-of-the-art text mining tools for this. The popular and widely used tf-idf [9][8], finds important words in a document that are otherwise rare in other documents. For a given set of pre-identified words, it finds the tf- word frequency in the document. This term frequency for each word is then divided by the frequency of the word in the entire corpus. One drawback of tf-idf, is that it does

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

not reduce the size of the feature set substantially which slows down the computation in case where there is a huge dataset. Latent Semantic Indexing uses singular value decomposition on this and creates a feature vector which captures the maximum variance. Another advancement to this was aspect modeling, or probabilistic LSI (pLSI) suggested by Hoffman.

Latent Dirichlet Allocation(LDA),[1][8] was another breakthrough in the area of text mining. This finds out the latent topics in the corpus, where each topic is a set of similar words. Each document is a probabilistic mixture of these topics. LDA is known to give better results compared to the previously discussed techniques.

Most of these techniques are computationally intensive and so we ventured into Vowpal Wabbit which is an open source, fast machine learning library developed at Yahoo! Research and currently under Microsoft Research [5]. We exploited the hashability property of VW library to get better results.

Boosted Regression Tree, recursively partitions the data and models the data for each partition [2][6]. It is also extensively used in many data mining applications. This is known to work better than linear regression, in case of certain kind of datasets.

3. DATASET

3.1 Basic Statistics and Properties

We used Yelp dataset from the sixth round of Yelp data challenge for this project. The dataset provides the following information:

1. Business (61,184 businesses)
2. Review (1,569,264 reviews)
3. Users (366,715 users)
4. Check-in (Check -in information for 45,166 businesses)
5. Tips (495,107 tips)

For this project, only Business and Review data were used. Further, among the businesses only restaurant data were considered.This reduced the number of businesses and reviews to 21,882 and 978,481 respectively.

As the size of the data was large, we used only the data of 5 cities with highest number of reviews. This reduced the number of reviews for analysis to 685,007. Table 1 depicts distribution of reviews over these cities.

Table 1: Distribution of reviews across cities

City	No. of Reviews
Las Vegas	365739
Phoenix	139,343
Scottsdale	77,403
Charlotte	59,058
Pittsburg	43,464

3.2 Exploratory Data Analysis

To develop a deeper insight into how the sentiment from the review text varied over geographical location, time, etc. we carried out extensive exploratory analysis on our data. Few snapshots and interesting findings are below.

3.2.1 Rating Distribution

Figure 3 shows how the ratings are distributed across the data. It can be seen that more people tend to give more positive reviews than negative reviews.

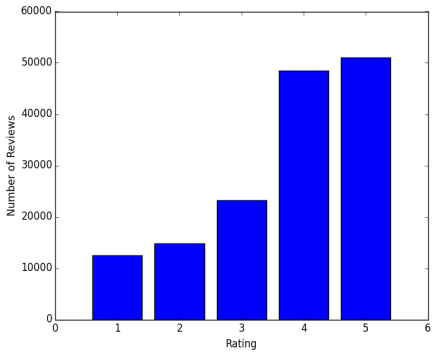


Figure 3: Plot of number of reviews vs rating.

3.2.2 Wordclouds

The word clouds based on word frequency for review texts rated 1 and 5 have been shown in Figure 2. The unigram models based on word frequency alone, for rating 1 and 5, do not give much information as the same terms are repeated for both. From bigram word cloud for rating 1 we can see that, dissatisfied customers stress on long wait time (10, 15, 20 minutes) and terrible customer service much more and more often than the taste of food. While reviews for rating 5 clearly talk about the taste (great food) and great service. Most reviewers also specifically mention whether they would come back or not. Trigram model does not give much information over the bigram model. We observe almost a similar phenomenon when we plot word clouds for other cities.

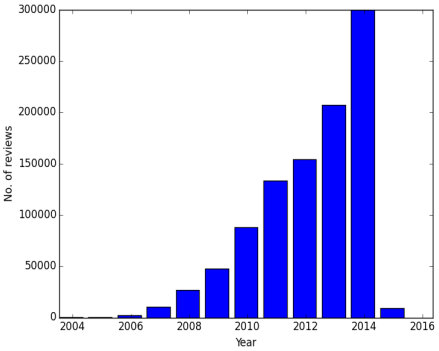


Figure 4: Plot of number of reviews vs year.

3.2.3 Temporal Factor Yearly

1) Number of reviews vs Year: Yelp was founded in 2004 and by 2006 it had a significant number of reviews. Since, then we see a constant increase in the number of reviews which can be seen in Figure 4. The next two plots provide some more interesting information about how temporal factors affect the ratings.

2) Increase in reviews over the previous year: Figure 5 shows the percentage increase in the number of reviews

(a) Rating 1 Unigrams (b) Rating 5 Unigrams (c) Rating 1 Bigrams

(d) Rating 5 Bigrams (e) Rating 1 Trigrams (f) Rating 5 Trigrams

Figure 2: Unigram, Bigram and Trigram word clouds for Rating 1 and 5.

over the previous year. Since 2009 we see almost a constant percentage increase in reviews, whereas before 2009 we see that the rate of increase increases every year. This year also marks the launch of yelp mobile app (Dec 2008, Source: Wikipedia) which did lead to substantial increase in yelp users, post which the rate of increase had saturated.

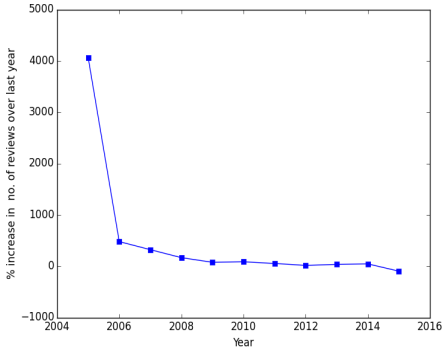


Figure 5: Plot of percentage increase in the number of reviews over the last year.

3) Average rating vs Year: The above hypothesis was further strengthened when we plotted the graph for average rating against the year (shown in Figure 6). The average ratings for years before 2009 are much different than those after.

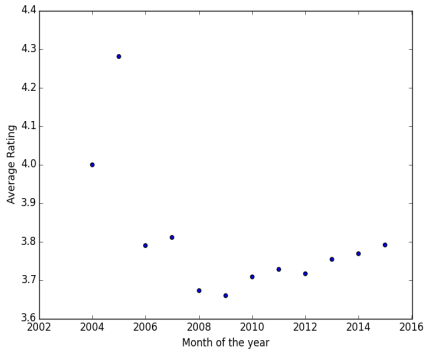


Figure 6: Plot of average user rating vs year.

3.2.4 *Temporal Factor Monthly*

1) Average rating vs month of the year: Figure 7 shows that the average rating and the number of reviews for the months of June, July and August are higher than the other months, while that for December is the lowest. Maybe, people are more critical during December and generous during the summer. This could be also as more people take vacation and eat out during summer.

3.2.5 *Geographical Factor*

Average rating vs city: Figure 8 shows that the average rating for the chosen cities for our analysis is also different. This motivated us to analyze how sentiments of people change across different cities. This is discussed more elaborately in Section 5 and 6.

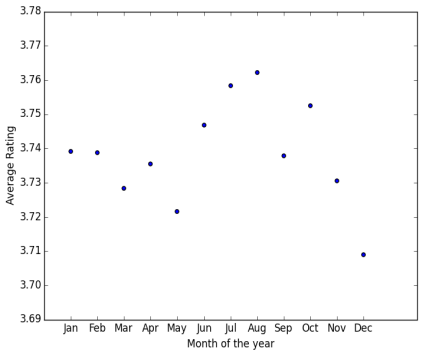


Figure 7: Plot of average user rating vs month of the year.

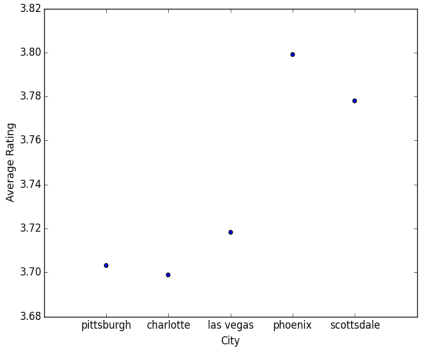


Figure 8: Plot of average user rating vs city.

4. PREDICTIVE TASK

The literature survey helped us in identifying the fact that the review text contains rich information which can be used to predict the rating. From EDA we identified a contrast in user rating patterns before and after 2009. We also found that the geographical location has an impact in the sentiments of people in rating the restaurants as the average rating is not uniform across all cities. Based on above observations, the following predictive tasks are formulated:

1. Prediction of user rating from the review text.
2. Sentiment analysis of users across the geographical locations.
3. Sentiment analysis of users over time.
4. Group users with similar tastes together.

4.1 Machine Learning Problem

The mentioned problems can be solved by using classification or regression. We found that the regression is better suited for the problem when compared to the classification. For example, if the rating is predicted as 2 instead of 1 and if it is predicted as 5 instead of 1, the errors in both cases are same by classification problem. But the prediction 2 must be penalized less than that of 5. This motivated us to choose regression.

In the final task, we tried to learn user's tastes and preferences from their reviews and perform clustering to group users with similar interests together.

4.2 Evaluation Metrics

4.2.1 Regression

Baseline.

Every rating is predicted as the average rating of all the reviews in the dataset.

Metric.

The task in regression is to reduce the Mean Squared Error (MSE) of the predicted values using the selected model, when compared to the baselines by a huge margin.

4.2.2 Clustering

Baseline.

We compared our model against a NULL model where users are grouped randomly.

Metric.

We defined a metric to compute the error for this model which is the mean squared difference of the ratings given by two different users within the same cluster to the same restaurant. Mathematical formulation for this metric is given below:

$$Error = \frac{\sum_{c \in C} \sum_{u_{i,c}, u_{j,c}}^{i \neq j} \sum_{r_{i,j}} (R(u_{i,c}, r_{i,j}) - R(u_{j,c}, r_{i,j}))^2}{\sum_{c \in C} \sum_{u_{i,c}, u_{j,c}}^{i \neq j} \sum_{r_{i,j}} 1} \quad (1)$$

where C is the set of all clusters, $u_{i,c}$ is the i^{th} user in cluster c, $r_{i,j}$ is the restaurant reviewed by both $u_{i,c}$ and $u_{j,c}$ and $R(i, r)$ is the rating given by user i to restaurant r.

We evaluate our model with the baseline using the ratio r defined below. The value of this ratio will be 0 for the optimal and 1 for the simplest model (NULL model).

$$r = \frac{Error_{clustering}}{Error_{NULL}} \quad (2)$$

4.3 Features

We performed cleaning of the reviews by removing stopwords and punctuation from the text. After this we considered 5 different featurization techniques to train the system using different regression models which will be discussed in later sections.

1. Term frequency of top 1000 unigrams.
2. Term frequency of top 1000 bigrams.
3. 12 bit Vowpal Wabbit with unigrams.
4. 12 bit Vowpal Wabbit with bigrams.
5. Latent Dirichlet Allocation: Due to complexity of the algorithm, only 5000 reviews were used to learn topics in the reviews.

For the clustering task, we have used LDA with 100 topics as features.

5. MODEL

Many trials have been made in order to come up with an appropriate model for the task mentioned in the previous section.

5.1 Linear Regression

Now the task is to choose the appropriate features and the model. We began with linear regression and considered term frequency of top 1000 unigrams in every review to train the system and the MSE has been noted. As the data is huge for all the cities combined, we have used the reviews in single city for the training. The system is found to work better than the classification model and we tried to move further in this direction. The linear regression model with bigram bag of words and LDA topic modelling is found to have more MSE compared to the unigram model. Considering the complexity and accuracy of the model, we found that the linear regression model with the unigram is found to outperform bigram model and LDA.

5.2 Boosted Decision Tree Regression

Here, we found an interesting model of regression after going through literature, which is Boosted regression decision tree. Vowpal Wabbit with unigrams is used for feature construction. Boosted decision tree regression model along with Vowpal Wabbit with unigram features is found to give the best accuracy and reduced the MSE significantly when compared to the linear regression model mentioned above and hence it is picked for solving the rating prediction problem. For the same classifier we used bigrams as features and MSE is found to increase. It is concluded from our experiments that Boosted Decision Tree Regression with Vowpal Wabbit (unigram) features is found to have the least MSE which helped us to freeze this model and work further on geographical and temporal variations of sentiment analysis.

5.3 Geographical and Temporal Analysis

Based on our initial data analysis, we intend to study how sentiments of people change with geographical location. For this, different regression models were trained for each of the five cities and analyzed the top positive and negative words in every city.

Further, we wanted to analyze how the sentiments of people vary with time. More specifically, we wanted to see if there was any change in the way people give reviews to restaurants after Yelp mobile app was launched in December 2009. Similar to geographical analysis, we trained two different regression models for two different time frames and analyzed the top positive and negative words for a city.

5.4 User Clustering using K-Means

From the dataset of each city, we sorted the restaurants on the basis of the number of reviews they have got and chose the top 30 restaurants. We use these restaurants and the users who reviewed them for our analysis. For every two users who reviewed at least one common restaurant, we measured their similarity as,

$$S(u_1, u_2) = \frac{\sum_{r \in R_{u_1,u_2}} Cosine(u_{1,r}, u_{2,r})}{\sum_{r \in R_{u_1,u_2}} 1} \quad (3)$$

where R_{u_1,u_2} is the set of all restaurants reviewed by both u_1 and u_2, $Cosine(a, b)$ is the cosine similarity between vector a and b, and $u_{1,r}$ is the feature vector computed using LDA for the review of u_1 for restaurant r.

Next, we used the k-means clustering algorithm to group users into clusters based on their similarity. We restricted the number of clusters to two for our analysis.

6. RESULTS

The results of our experiments are reported below. In all tasks the data was partitioned into 80-10-10 ratio and the parameters of the model were learned using random grid search on 10% validation set.

6.1 Rating Prediction

Considering the large size of the data, we performed this task on the data of individual cities. Table 2 reports MSE of different models used with different feature extraction techniques for the city of Phoenix. It can be observed that every model outperforms the baseline quite comfortably. Among all models, Boosted Decision Tree Regression with Vowpal Wabbit (unigrams) yields the least MSE.

Table 2: Rating prediction for Phoenix using different models. Refer Appendix for abbreviations.

Model	Feature	MSE
Baseline		1.551783
LR	BOW + Unigram	0.832204
LR	BOW + Bigram	0.892364
LR	VW + Unigram	0.837587
LR	VW + Bigram	0.895167
LR	LDA K=50	1.378367
LR	LDA K=100	1.219397
LR	LDA K=200	1.295201
BTR	BOW + Unigram	0.774963
BTR	VW + Unigram	**0.667232**
BTR	VW + Bigram	0.743602

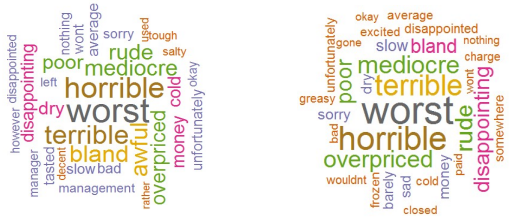
6.2 Geographical Sentiment Analysis

Table 3 compares the MSE of linear regression with bag of words model with boosted decision tree with vowpal wabbit for all five cities being considered. We trained different models for each city. For every city, boosted decision tree regression with vowpal wabbit provides the significantly lower MSE value.

Table 3: Rating prediction for different cities. Values shown are MSE for different models. Refer Appendix for abbreviations.

City	Baseline	LR + BOW	BTR + VW
Charlotte	1.412087673	0.784653	**0.662986778**
Pittsburgh	1.429535	0.793964	**0.676914272**
Phoenix	1.551783	0.832204	**0.667232487**
Scottsdale	1.561418	0.807391	**0.648467826**
Las Vegas	1.609975	0.852307	**0.648975248**

Figure 9 shows a comparison of top 35 negative words used in the review between Charlotte and Las Vegas. Though most of the words looks same across the two cities, one interesting observation we drew from it was that the word *decent* is observed as a highly negative word in Charlotte (figure 9a), whereas the same word is a positive word (not shown in the figure) in Las Vegas. This shows that there is a difference in the way the same words are used in different cities. Similar observations were found for other cities, which are not shown in this paper.



(a) Top negative words for Charlotte.

(b) Top negative words for Las Vegas.

Figure 9: Comparison of negative sentiments for Charlotte and Las Vegas through word cloud.

6.3 Temporal Sentiment Analysis

Table 4 shows the MSE of rating prediction before and after 2009 for Las Vegas. We have shown results only for one city because the other cities had less than 10k reviews before 2009. The high MSE for reviews before 2009 could be due to less training data (13k). However, we observe that the MSE after 2009 is 0.46% lower than the MSE obtained for Las Vegas in Table 3. This justifies our hypothesis behind having two different models for different time frames.

Table 4: Rating prediction for Las Vegas before and after 2009. Values shown are MSE for different models. Refer Appendix for abbreviations.

Year	Baseline	LR + BOW	BTR + VW
Before 2009	1.30867	0.819501	**0.7333986937**
After 2009	1.623485	0.854151	**0.6459770905**

Similar to the geographical sentiment analysis, we compared the positive and negative sentiments of the people in reviews between two time frames. Figure 10 shows word cloud of top 35 positive words used in reviews during two time frames. Unlike the geographical analysis, we did not observed anything really interesting. However, it could be seen that some words like *normally* and *attentive*, which are not so strong positive adjectives are observed as highly positive before 2009 whereas words after 2009 are mostly strong positive adjectives.

(a) Top positive words before 2009.

(b) Top positive words after 2009.

Figure 10: Comparison of positive sentiments for Las Vegas before and after 2009.

6.4 User Clustering

For this task, we used the reviews of 30 most reviewed restaurants in Phoenix. We divided the users who reviewed these restaurants into two clusters using spherical K-Means clustering. We obtained two clusters of size **4011** and **3589** and the ratio defined in Section 4.2.2 as **0.9113**. Though we did not see a significant improvement over the NULL model but there is certainly an improvement over the baseline.

7. CONCLUSIONS

In this project, we studied various models in order to predict the user rating from review text. We also analyzed how sentiments of users change with geographical location and the year of rating. It is discovered that Boosted Decision Tree Regression model with Vowpal Wabbit and unigram features is found to give the least MSE. In addition, we grouped users based on similarity of their reviews for similar restaurants. Although, we formed cluster of users using k-means and evaluated it on a defined metric, we did not use it to predict the rating. We believe that when used as a feature it could improve the accuracy of the prediction task.

8. ACKNOWLEDGMENTS

We would like to thank Prof. Julian McAuley for his guidance and suggestions.

9. REFERENCES

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[2] J. Elith, J. R. Leathwick, and T. Hastie. A working guide to boosted regression trees. *Journal of Animal Ecology*, 77(4):802–813, 2008.

[3] M. Fan and M. Khademi. Predicting a business star in yelp from its reviews text alone. *arXiv preprint arXiv:1401.0864*, 2014.

[4] G. Ganu, N. Elhadad, and A. Marian. Beyond the stars: Improving rating predictions using review text content. In *WebDB*, volume 9, pages 1–6. Citeseer, 2009.

[5] J. Langford, L. Li, and A. Strehl. Vowpal wabbit online learning project, 2007.

[6] W.-Y. Loh. Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):14–23, 2011.

[7] J. McAuley and J. Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172. ACM, 2013.

[8] J. Ramos. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, 2003.

[9] Stanford.edu. tf-idf. http://nlp.stanford.edu/IR-book/html/htmledition/tf-idf-weighting-1.html/.

[10] J. Tang, H. Gao, X. Hu, and H. Liu. Context-aware review helpfulness rating prediction. In *Proceedings of the 7th ACM Conference on Recommender Systems*, pages 1–8. ACM, 2013.

APPENDIX

A. ABBREVIATIONS

LR: Linear Regression; **BOW:** Bag of Words; **BTR:** Boosted Decision Tree Regression; **VW:** Vowpal Wabbit.