Data Mining Yelp Data - Predicting rating stars from review text

Rakesh Chada
Stony Brook University
rchada@cs.stonybrook.edu

Chetan Naik
Stony Brook University
cnaik@cs.stonybrook.edu

ABSTRACT
The majority of the online reviews are written in free-text format. It is often useful to have a measure which summarizes the content of the review. One such measure can be sentiment which expresses the polarity (positive/negative) of the review. However, a more granular classification of sentiment, such as rating stars, would be more advantageous and would help the user form a better opinion. In this project, we propose an approach which involves a combination of topic modeling and sentiment analysis to achieve this objective and thereby help predict the rating stars.

Keywords
Yelp, Topic Modeling, Sentiment Analysis, Latent Dirichlet Allocation, Non-negative Matrix Factorization, Naive Bayes Classifier, tf-idf

1. INTRODUCTION
The goal of our project is to predict a review's star rating given just the review's text. To understand the importance of this, let's consider the task of predicting the rating of a newly released movie. The audience who had watched the movie would have expressed their opinion through tweets. So, in this scenario, it would be useful to have a model which takes all such tweets as an input and predicts the rating. This task of predicting star rating from text is posed as a sub-problem in Yelp Dataset Challenge [1].

2. PRIOR WORK
The key aspect of our proposal involves topic modeling and we rely on two state-of-the-art techniques to achieve this task.
The first technique we used is Latent Dirichlet Allocation (LDA)[2] proposed by Blei in 2003 and the second technique we used is Non-negative Matrix Factorization (NMF)[3]. In Blei et al.[2], the author proposes a novel approach for topic modeling which improves upon the previous state-of-the-art technique called Probabilistic Latent Semantic Indexing[4] (PLSI). The topics are assumed to be drawn out of a Dirichlet distribution and it solves the problem of over-fitting faced by PLSI. One successful attempt to use NMF for topic modeling has been demonstrated in Arora et al.[3]. It was shown that NMF helps in determining correlation between the topics and hence it generalizes to more realistic models. There also has been significant research in the area of sentiment analysis. A successful application of machine learning techniques to perform sentiment analysis can be observed in Pang et al.[5]. Here, classifiers such as Naive Bayes and Support Vector Machines were used to determine the sentiment.

3. DATA COLLECTION
We used the Yelp Academic Dataset[1] which consists of 1,125,458 reviews, 320,002 business attributes and 252,898 users. This data is in JSON format. The dataset contains following information:

Name	Attributes
Business	Business Name, Id, Category, Location, etc.
User	Name, Review Count, Friends, Votes, etc.
Review	Date, Business, Stars, Text, etc.

However, we have focused only on restaurant reviews as they account for major part of the dataset. The restaurant review dataset contains 706,692 entries and 135 columns which amounts to around 1GB of data.

4. MODEL
4.1 Baseline Model
An interesting observation that resulted from a histogram plot of review ratings has led us to build a baseline model that is based on average rating. The average rating of all the restaurant reviews in our data set is 3.7 and it is rounded off to 4 and used as the baseline prediction. The accuracy of the baseline model is demonstrated in the table below.

	precision	recall	f1-score
Baseline	0.11	0.33	0.16

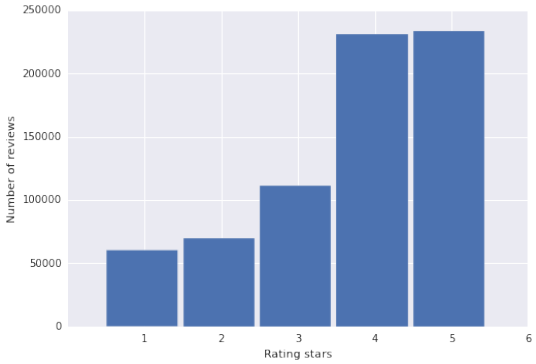

Figure 1: The plot shows a histogram of rating stars. We can observe that most of the reviews are rated 4 and 5 stars.

4.2 Advanced Models

Once we had the baseline model, we built several advanced models that helped predict the rating stars. All of these models have a common underlying theme which is extracting key features of a review to help predict the sentiment. The difference in these models is the type of features that are used to train the model. The first model uses term frequencies, the second uses topics and the further ones use topics combined with the sentiment. This workflow which we have used is better visualized in the flowchart below:

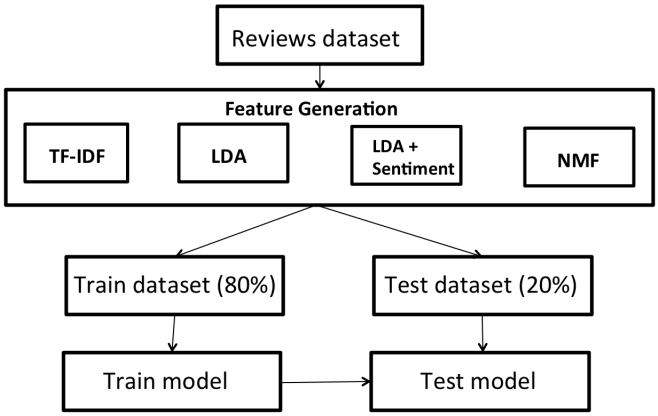

Figure 2: Flow chart representing the design of our workflow

As illustrated in the above figure, features from the review dataset are extracted using several techniques such as TF-IDF, Latent Dirichlet Allocation (LDA) and Non-negative Matrix Factorization (NMF). After the feature extraction, the data is split randomly into train data (80%) and test data (20%). The model trained using these features is then evaluated on the test data.

4.3 Term Frequency Classifier

This approach uses word frequencies as features to train the model. Intuitively, word frequency can be thought of as an indicator of the sentiment. For example, the fact that 'amazing' is repeated twice in the review 'Amazing food and amazing service!!' indicates that the review is oriented towards positive sentiment. Once we train the model using term frequencies, we pass those features to classifiers such as Naive Bayes and Logistic Regression to predict the sentiment. For k-nearest neighbors, we used k = 5 for the classification. Results are illustrated in the table below:

	precision	recall	f1-score
Logistic Regression	0.57	0.58	0.55
Multinomial Naive Bayes	0.51	0.52	0.44
Nearest Neighbors	0.48	0.50	0.49

The error distribution for each of the classifiers is shown below.

In the plots below, the green circle represents the mean error and the red circle represents the median error. The two ends of the green line represent the standard deviation and the two ends of the red line represent the central 68 percentage of the data.

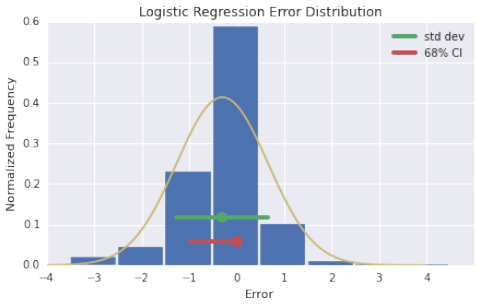

Figure 3: Probability distribution of error for logistic regression using term frequencies as features

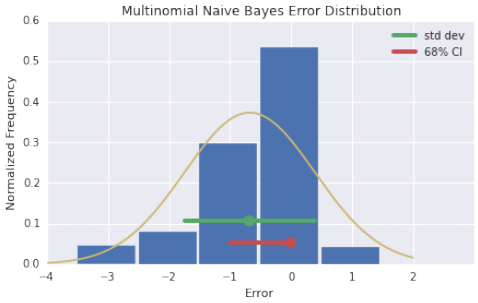

Figure 4: Probability distribution of error for Multinomial Naive Bayes Classifier using term frequencies as features

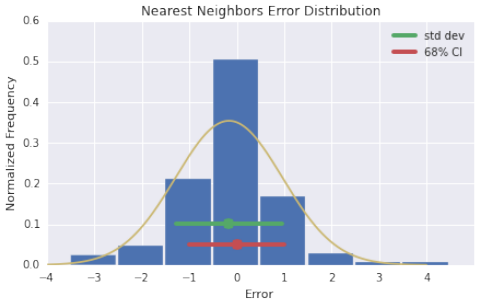

Figure 5: Probability distribution of error for Nearest Neighbors Model using term frequencies as features

4.4 Topic based Classifier - LDA

The previous model considers all the words and their frequencies as training features. However, it might be inefficient to do such a task when the data is huge and also it might affect the accuracy of prediction when the features span a wide range of words. So, we extracted only key features of a review, called topics, and used them as training features. To extract topics, we used Latent Dirichlet Allocation[2] proposed by Blei. The results of this model are demonstrated below:

	precision	recall	f1-score
AdaBoost	0.39	0.45	0.39
Logistic Regression	0.33	0.45	0.33
Multinomial Naive Bayes	0.20	0.45	0.28

The error distribution for each of the classifiers is shown below.

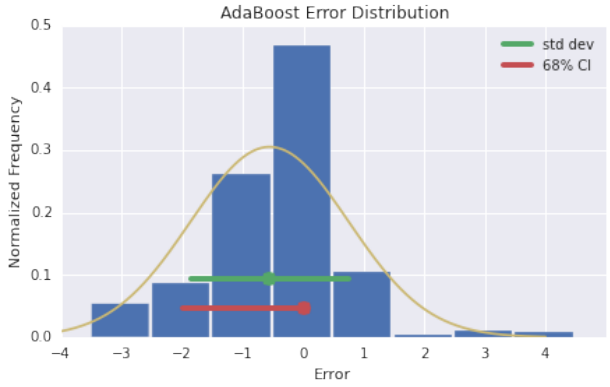

Figure 6: Probability distribution of error for Adaboost Model using topics as features

Figure 7: Probability distribution of error for Logistic Regression Model using topics as features

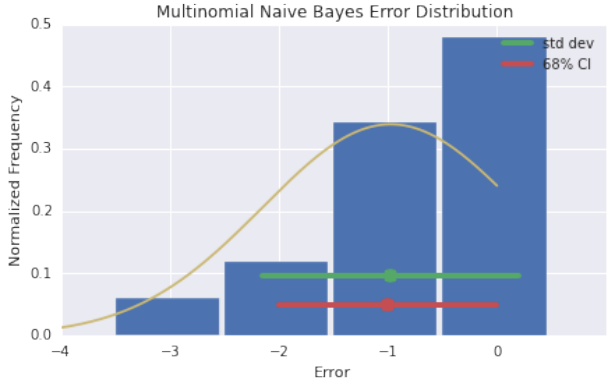

Figure 8: Probability distribution of error for Multinomial Naive Bayes Model using topics as features
From the above three plots, we can infer that mean and median in the error distribution are closer to zero in case of AdaBoost model and closer to -1 in case of Logistic Regression and Multinomial NaÃŕve Bayes Classifier.

4.5 Topic and Sentiment based Classifier - LDA

As observed in the results above, using topics as features to train the classifier resulted in a lower accuracy than using term frequencies. This could be due to the fact topics don't have a sentiment associated with them. For example, consider the following two reviews:

Review 1: "This place has great ambience."
Review 2: "This restaurant doesn't provide valet parking."

The first review talks about ambience in a positive manner and the second review talks about parking in a negative sense but the topics extracted from these reviews would just be 'ambience' and 'parking'. They don't capture the essence of the sentiment and thereby would not serve useful as features to determine star rating.

This led us to the idea of adding sentiment as a feature along with topics to train the model. We used Naive Bayes Classifier to extract the sentiment and the results of the sentiment prediction are as follows:

	precision	recall	f1-score
Logistic Regression	0.88	0.96	0.92
Multinomial Naive Bayes	0.82	0.99	0.90
Nearest Neighbors	0.84	0.96	0.90

The extracted sentiment, along with the topics, are passed to the classifier and the results obtained are demonstrated below:

	precision	recall	f1-score
AdaBoost	0.46	0.49	0.46
Logistic Regression	0.44	0.50	0.40
Multinomial Naive Bayes	0.20	0.45	0.28

The error distribution for each of the classifiers is shown below.

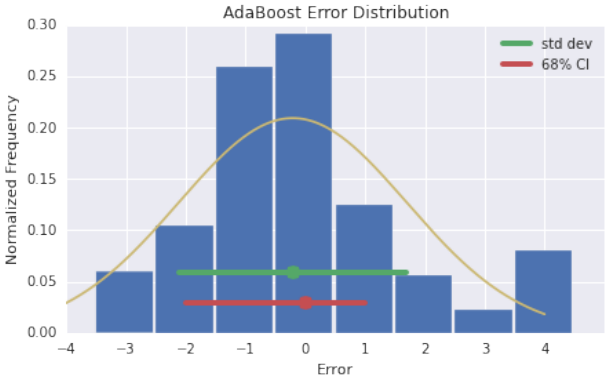

Figure 9: Probability distribution of error for Adaboost Model using topics (LDA) and sentiment as features

Figure 10: Probability distribution of error for Logistic Regression Model using topics (LDA) and sentiment as features

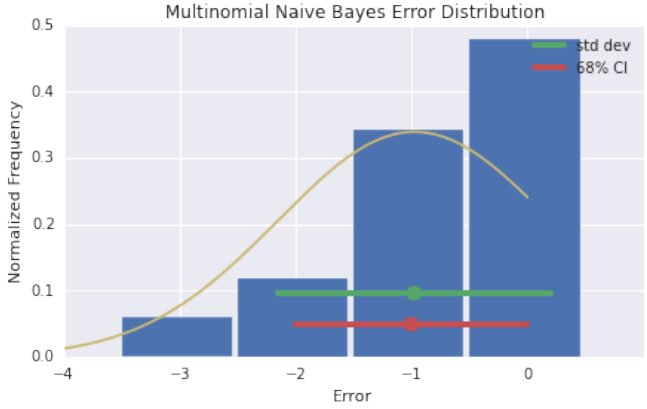

Figure 11: Probability distribution of error for Multinomial Naive Bayes Model using topics (LDA) and sentiment as features

From the above three plots, it can be inferred that the accuracy of the predictions, in case of AdaBoost and Logistic Regression, has increased compared to the previous model that is just based on topics. However the Multinomial NaÃŕve Bayes model didn't show any improvement.

4.6 Topic and Sentiment based Classifier - NMF

Latent Dirichlet Allocation (LDA) is a probabilistic model and hence there is no single representation of the corpus. This led us to evaluate our model by using topics generated from a deterministic model such as Non-negative Matrix Factorization. These topics, along with the sentiment extracted using Naive Bayes classifier, were passed as features to train the model. The results obtained are demonstrated below:

	precision	recall	f1-score
AdaBoost	0.59	0.61	0.59
Logistic Regression	0.60	0.61	0.58
Multinomial Naive Bayes	0.40	0.49	0.39

The error distribution for each of the classifiers is shown below.

Figure 12: Probability distribution of error for Adaboost Model using topics (NMF) and sentiment as features

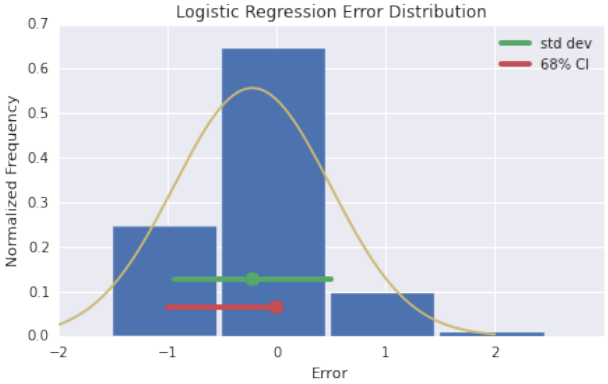

Figure 13: Probability distribution of error for Logistic Regression Model using topics (NMF) and sentiment as features

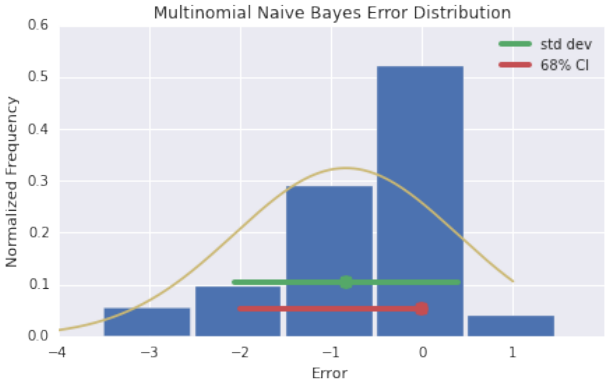

Figure 14: Probability distribution of error for Multinomial Naive Bayes Model using topics (NMF) and sentiment as features

The error distribution, with mean and median close to zero, behaves similar to that of a standard normal/Gaussian distribution in the case of AdaBoost and Logistic Regression. The summary statistics of all our models is represented in the table below:

Model	Classifier	Precision	Recall	f1-Score
Baseline	-	0.11	0.33	0.16
tf-idf	LR	0.56	0.57	0.55
	NB	0.51	0.51	0.43
LDA	AB	0.38	0.45	0.38
	LR	0.33	0.44	0.32
	NB	0.20	0.44	0.27
LDA + Sentiment	AB	0.45	0.48	0.45
	LR	0.44	0.49	0.39
	NB	0.20	0.44	0.27
NMF + Sentiment	AB	0.59	0.60	0.59
	LR	0.59	0.61	0.58
	NB	0.40	0.48	0.38

Note:
LR - Logistic Regression
NB - Multinomial Naive Bayes
AB - AdaBoost

It can be observed that NMF with an additional sentiment feature performs significantly better than the model based on LDA. It is also interesting to notice that tf-idf model when used with Logistic regression resulted in a similar level of accuracy as that of NMF with sentiment layer.

5. CONCLUSION

The motivation for this paper is to come up with a method to predict a review's star rating from its review text. This has applications in tasks such as information retrieval, opinion mining, text summarization and many other problems which involve large amount of textual data. In this report, we have discussed our approach which involved the combination of topic modeling and sentiment analysis to predict the star rating. Several feature extraction methods such as term frequency classifier, Latent Dirichlet Allocation (LDA) and Non-negative matrix factorization (NMF) have been compared and evaluated against our dataset. NMF with an added sentiment layer and tf-idf model produced results which are more accurate than LDA based model.

6. FUTURE WORK

In our model, we extracted topics and sentiment separately and used them as features while training. However, there are approaches such as the one proposed in Lin et al. [6] that extract both topics and sentiment simultaneously. The authors claim that the simultaneous extraction would improve the individual accuracy. So, it would be interesting to observe the results when topics and sentiment extracted using such joint approach are passed as features to train the model. Furthermore, it would be interesting to test the machine learning classifiers on the features extracted by fine tuning the parameters such as choosing optimal 'learning_rate' for AdaBoost model.

7. REFERENCES

[1] http://www.yelp.com/dataset_challenge

[2] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." the Journal of machine Learning research 3 (2003): 993-1022.

[3] Arora, Sanjeev, Rong Ge, and Ankur Moitra. "Learning topic models–going beyond svd." Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on. IEEE, 2012.

[4] Hofmann, Thomas. "Probabilistic latent semantic indexing" Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 1999.

[5] Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. "Thumbs up?: sentiment classification using machine learning techniques." Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. Association for Computational Linguistics, 2002.

[6] Lin, Chenghua, and Yulan He. "Joint sentiment/topic model for sentiment analysis." Proceedings of the 18th ACM conference on Information and knowledge management. ACM, 2009.