**DePaul University – College of Computing and Digital Media**

**ECT-584 Web Data Mining for Business Intelligence**

**Professor** - Jonathan Gemmell

**Final Project Report**

**Project Title:** A data mining approach to predict Yelp reviews rating stars

**Student:** Naga Venkateshwarlu Yadav Dokku

**2016 summer**

**Introduction**

The Yelp Company gathers reviews from users attending to restaurants, clinics and other services. Each review contains a star rating that is given by a user to the rated business. Combining all the ratings of all users gives a total rate for each business. The rating is given by the users and, in theory, it is based on the text of the review. Asking for a rating to a user that has already written a review is asking for duplicate information and storing duplicate information as well. Therefore, can we infer the rating of a review looking at the text written by the user and other metadata such as location, date, etc.? This way the criteria of ratings is unified and also the amount of information stored decreases which results in benefits for both the users and the Yelp Company. Yelp released a series of data sets as part of the Yelp Dataset Challenge contest. These data sets included anonymized information regarding business reviews, Yelp reviewers, check-in information and tips. The objective was to make this data available to (budding) data scientists and see what can be learned about reviews and business behaviors involving of businesses reviewed on Yelp. Here are some examples of topics we can find interesting:

Cultural Trends: By adding a diverse set of cities, we want participants to compare and contrast what makes a particular city different

Location Mining and Urban Planning: How much of a business' success is really just location, location, location? Do you see reviewers' behavior change when they travel?

Seasonal Trends: What about seasonal effects: Are there more reviews for sports bars on major game days and if so, could you predict that?

Natural Language Processing (NLP): How well can you guess a review's rating from its text alone? What are the most common positive and negative words used in our reviews?

Change points and Events: Can you detect when things change suddenly (i.e. a business coming under new management)? Can you see when a city starts going nuts over cronuts?

Social Graph Mining: Can you figure out who the trend setters and How much influence does my social circle have on my business choices and my ratings?

**Problem Statement**

Uncover hidden dimensions of ratings using both ratings and review text combined.

From the uncovered hidden dimensions, we can answer the following questions:

What does a particular user cares about regarding restaurants?

Which aspects should the restaurant improve in order to effectively increase the rating?

Which restaurant is the best for a particular user?

The aim of this project is to present the implementation of a prediction for star-rating in reviews based on learning from words in the existing rated reviews. In part of this project work, presented how to obtain the data used in this project as well as a description of all the phases the data has passed through to reach the final model, including data preprocessing, exploratory analysis, cleaning data and prediction modeling. And finally summarize and interpret the results obtained as well as the errors of the prediction model.

**Dataset**

Yelp has made a subset of its data publicly available in the context of the Yelp Dataset Challenge. Data is provided for users, tips, reviews checkins and businesses. 2.2M reviews and 591K tips by 552K users for 77K businesses,566K business attributes, e.g., hours, parking availability, ambience. Social network of 552K users for a total of 3.5M social edges. Aggregated check-ins over time for each of the 77K businesses. The data provided was exclusively in JSON format.

**Notes on the Dataset and Data preprocessing**

For doing a data analysis project downloaded the data available on the Yelp Dataset Challenge homepage.

Here’s the dataset link <https://www.yelp.com/dataset_challenge/dataset>

Each file is composed of a single object type, one json-object per-line.

Once downloaded the data and extracted we can observe the following data files:

[1] "yelp\_academic\_dataset\_business.json"

[2] "yelp\_academic\_dataset\_checkin.json"

[3] "yelp\_academic\_dataset\_review.json"

[4] "yelp\_academic\_dataset\_tip.json"

[5] "yelp\_academic\_dataset\_user.json" The data is stored as JSON files

Every object contains a 'type' field, which tells whether it is a business, a user, or a review. The data consists of Business objects that contain basic information about local businesses. The fields are as follows:



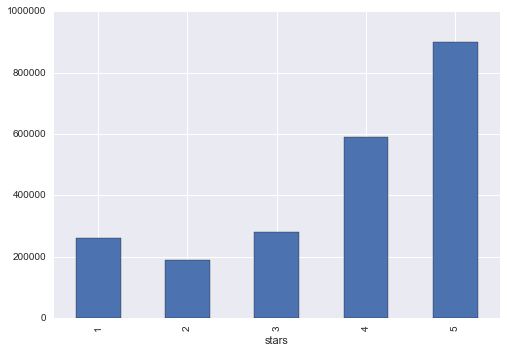
**Exploratory analysis**

The main goal of this section is to know more about the data, how it is organized, which relations exist and to see the correlation between reviews text and star-rating. Our data is split in several files, but the information is linked using ids. For example, the business dataset has a property called business\_id that identifies uniquely the business. At the same time, the review, checkin and tip datasets reference the business by the same business\_id property.

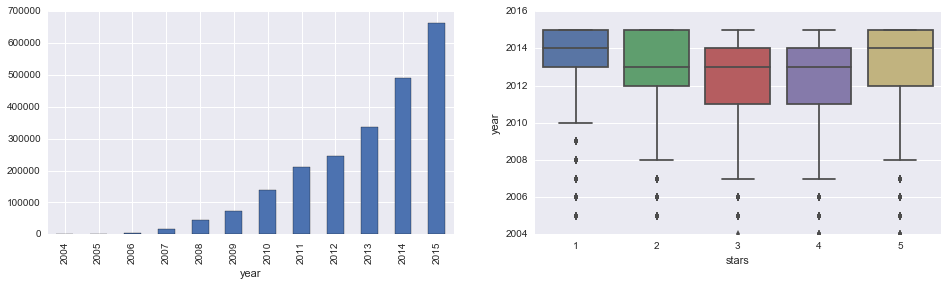
Cleaning data

Despite having transformed the original text-plain JSON files into lists, converted dataset from json from to csv. The python code link provided on the yelp website is used to convert the Yelp Dataset Challenge dataset from json format to csv. Therefore, so it can easily perform further analysis on this data.

Distribution of the Stars from reviews dataset

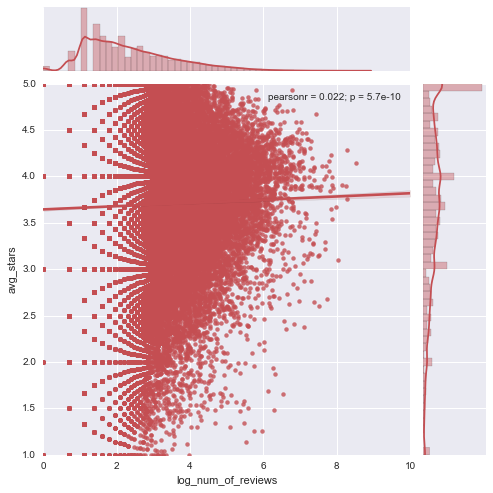
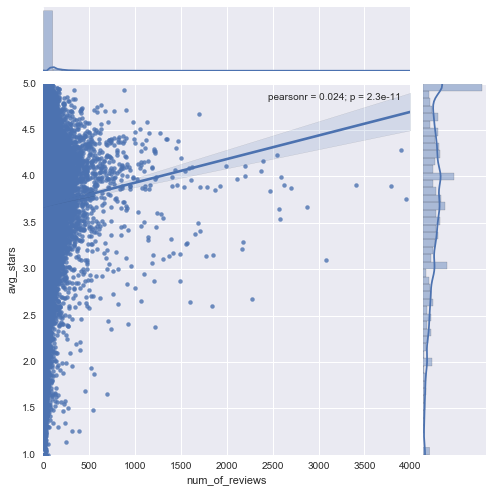


Analyzing the number of reviews and ratings per year



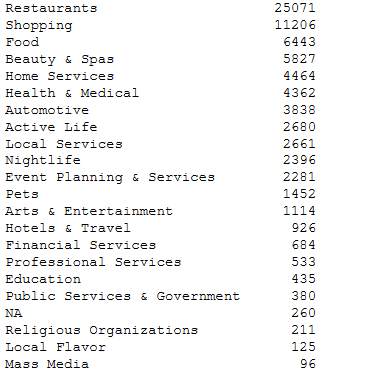
From above plots it looks the dataset consists of mostly recent reviews, and the star distributions doesn’t vary much over the years.

Now let's see if there's any relationship between the number of reviews a business receives and its average rating.

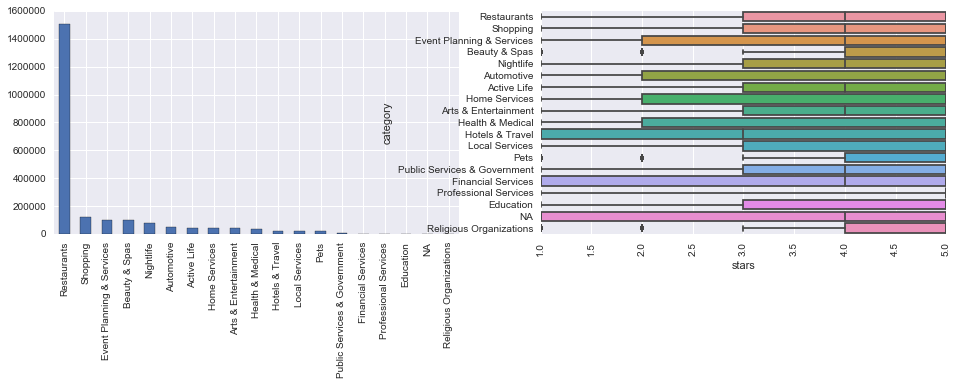


From the above two plots it’s seems to be an upward trend using the raw review count, but there is too much noise in the lower end to apply it reliably. When applying the log transformation, the trend disappears.

Frequencies of all categories for business dataset

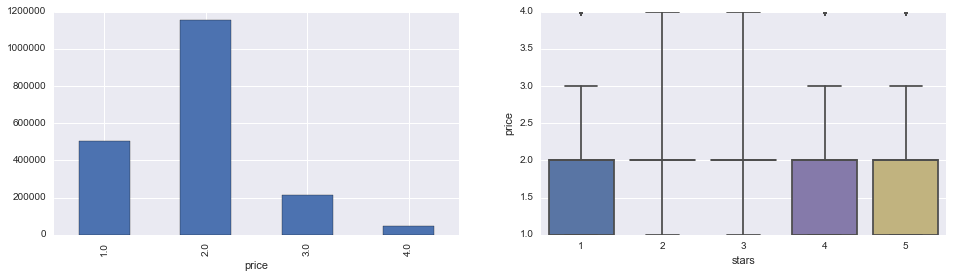


After merging reviews and business dataset let's take a look at how reviews and ratings are distributed across different categories:



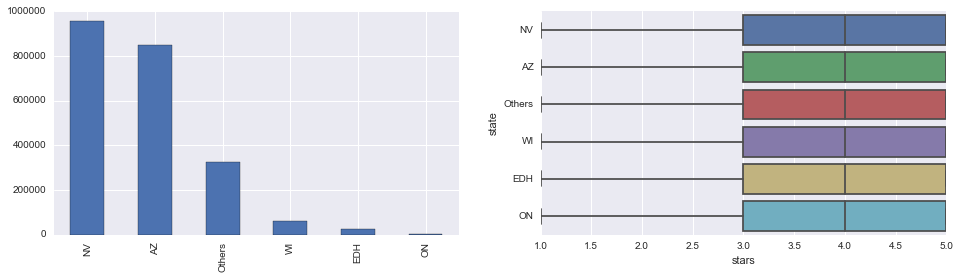
Not surprisingly, restaurants are the most popular subject for reviews and it greatly dwarfed the others. In terms of star ratings, it seems that, with the exception of the hotels and travel industry, people are mostly generous in reviewing businesses although some categories like financial services received a wide range of ratings.

Analyzing the number of reviews and ratings per price level



From the above plot it doesn't look very helpful in distinguishing the ratings.

Analyzing the number of reviews and ratings per state

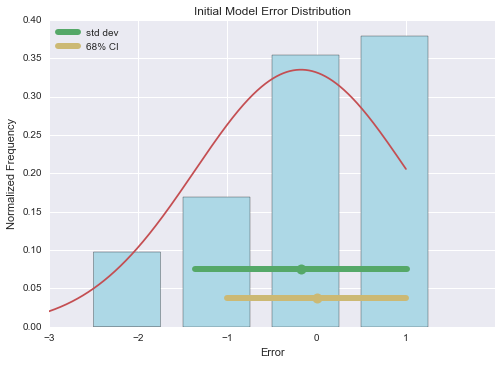
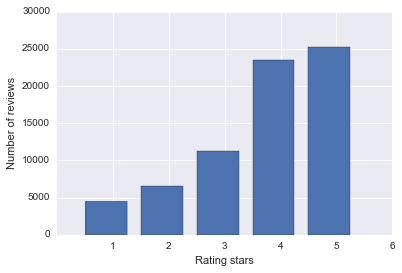


Models

Initial Model

An interesting observation that resulted from a histogram plot of review ratings has led me to build an initial model that is based on average rating. The average rating of all the restaurant reviews from data set is 3.7 and it is rounded off to 4 and used for an initial prediction. The accuracy obtained on initial model is 33%.





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

**Learning Models**

From the initial model, built few 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.

Reviews dataset

Feature Generation

Fe

NMF

LDA

TF-IDF

LDA+Sentiment

Test dataset (20%)

Train dataset (80%)

Train model

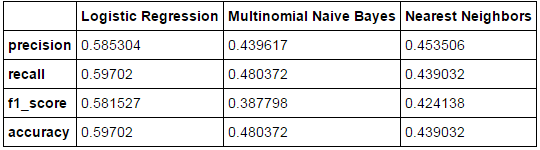
Test Model

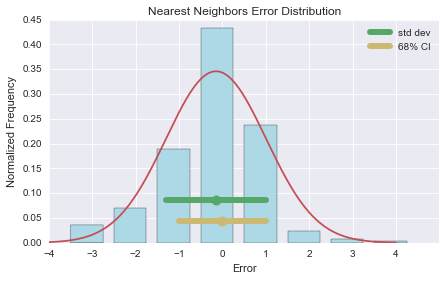
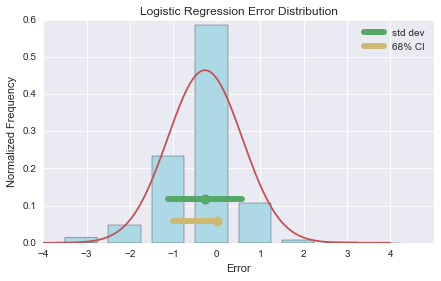
**Prediction modelling**

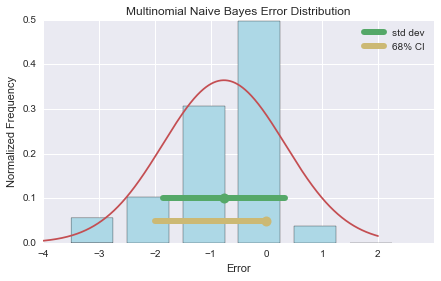
After all the preprocessing functions, the data set is merged on business\_id with business, reviews and users datasets. In this section, we will describe the process for building the prediction model using features. Base on the feature vectors generated above, applied different learning models to training samples. From 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.

**Term Frequency Classifier**

In this approach used 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 Naïve Bayes and Logistic Regression to predict the sentiment. For k-nearest neighbors, used k = 5 for the classification.



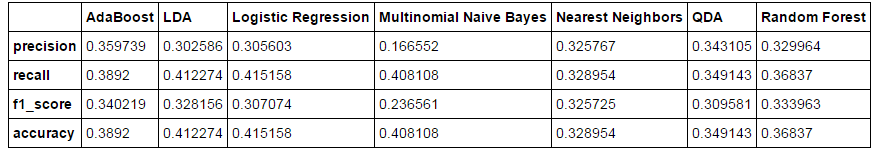


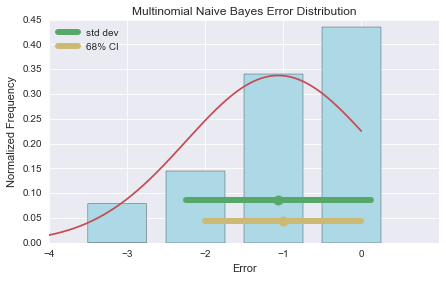
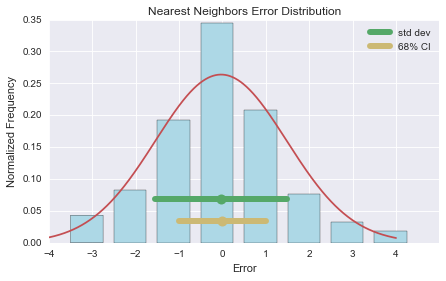
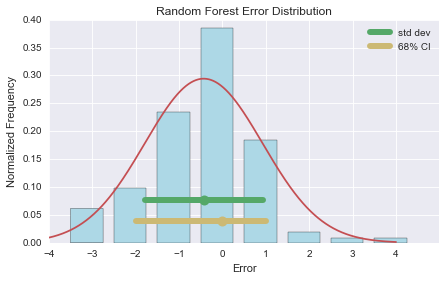
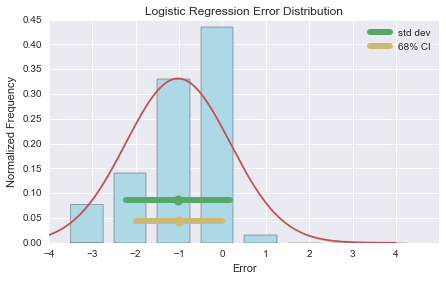
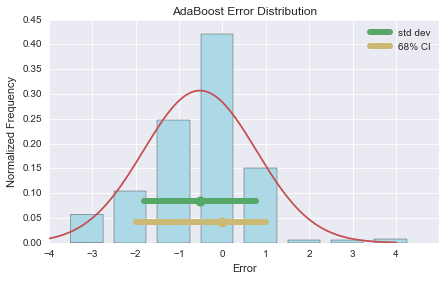


The above plots shows the error distribution for Logistic regression, Nearest Neighbors and Multinomial Naïve Bayes using TF as features. In the plots, the green circle represents the mean error and the gold circle represents the median error. The two ends of the green line represent the standard deviation and the two ends of the gold line represent the central 68 percentage of the data.

**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, used Latent Dirichlet Allocation proposed by Blei. With the topics, are passed to the classifier and the results obtained are demonstrated below:





The above plots shows the error distribution for each of the classifiers using topics as the 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, Random Forest and Nearest Neighbors models and closer to -1 in case of Logistic Regression and Multinomial Naive Bayes Classifier.

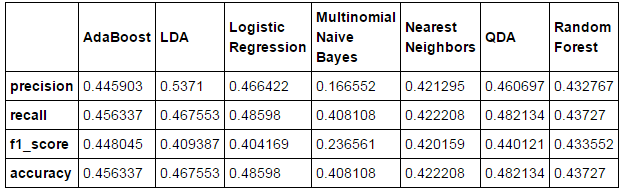
**Topic and Sentiment based Classifier – LDA**

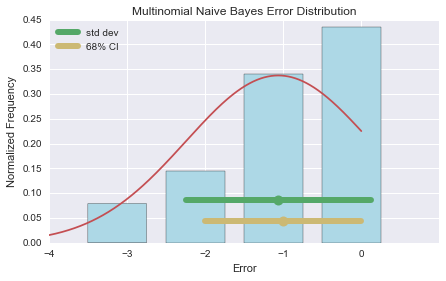
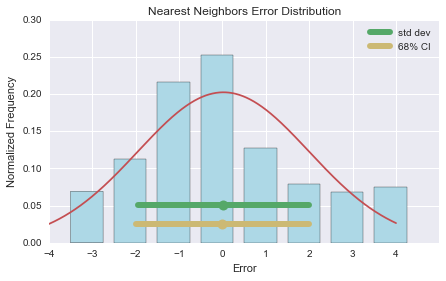
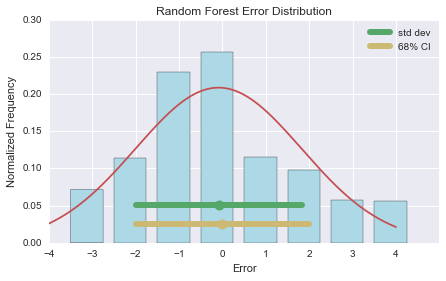
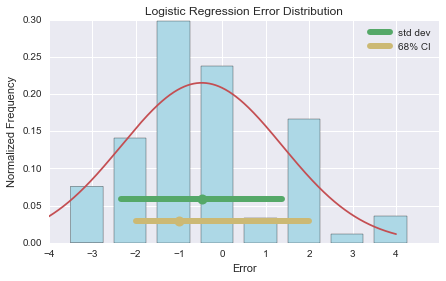
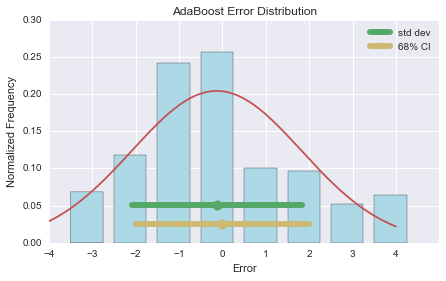
As observed in the from pervious LDA results, using topics as features to train the classifier resulted in a lower accuracy than using term frequencies. This could be due to the fact topics don’t have a sentiment associated with them. For example, consider the following two reviews:

Review 1: “Excellent food. Superb customer service.”

Review 2: “Yes this place is a little out dated.”

The first review talks about food in a positive manner and the second review talks about place in a negative sense but the topics extracted from these reviews would just be ‘Food’ and ‘Place’. They don’t capture the essence of the sentiment and thereby would not serve useful as features to determine star rating. So by adding sentiment as a feature along with topics to train the model. Naive Bayes Classifier is used to extract the sentiment and the results of the sentiment prediction are as follows:



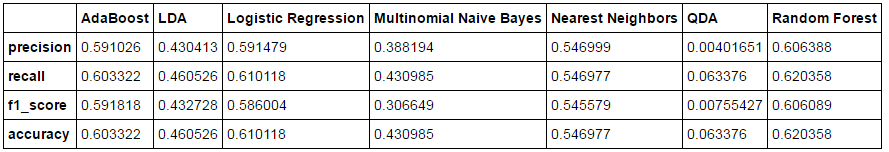


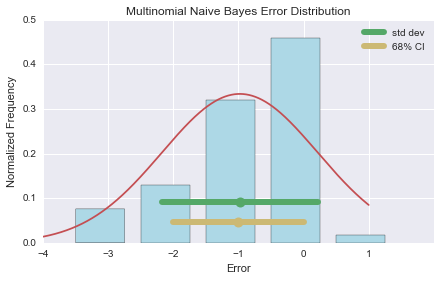
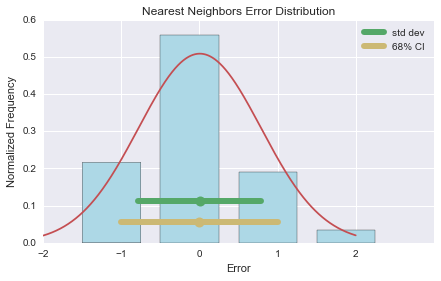
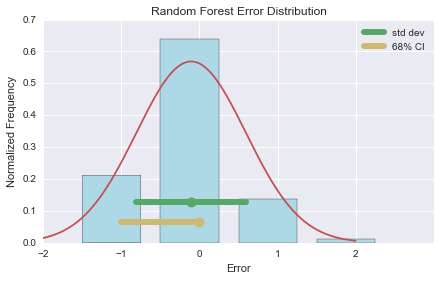
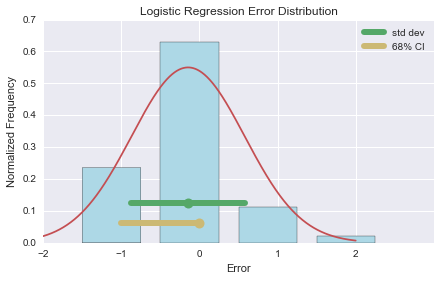
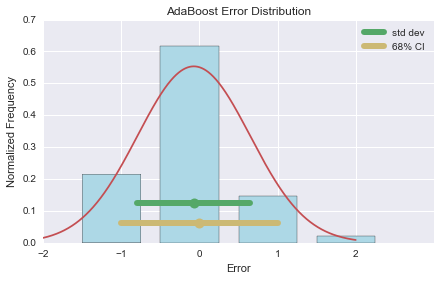
From the above three plots, it can be inferred that the accuracy of the predictions, in case of Nearest Neighbors and Logistic Regression, has increased compared to the previous model that is just based on topics. However the Multinomial Naïve Bayes didn’t show much improvement

**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 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 listed below:

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





The error distribution, with mean and median close to zero, behaves similar to that of a standard normal/Gaussian distribution in the case of Nearest Neighbors whereas for AdaBoost and Random Forest mean and median close to zero.

The summary statistics of all models is listed in the table below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Classifier | Precision | Recall | F1-Score | Accuracy |
| Initial | - | 0.11 | 0.33 | 0.16 | 0.33 |
| TF-IDF | LR | 0.59 | 0.60 | 0.58 | 0.60 |
| NB | 0.44 | 0.48 | 0.39 | 0.48 |
| LDA | AB | 0.36 | 0.39 | 0.34 | 0.39 |
| LR | 0.31 | 0.42 | 0.31 | 0.42 |
| NB | 0.17 | 0.41 | 0.23 | 0.41 |
| LDA+Sentiment | AB | 0.45 | 0.46 | 0.45 | 0.46 |
| LR | 0.47 | 0.49 | 0.40 | 0.49 |
| NB | 0.17 | 0.41 | 0.24 | 0.41 |
| NMF+Sentiment | AB | 0.60 | 0.60 | 0.59 | 0.60 |
| LR | 0.60 | 0.61 | 0.59 | 0.61 |
| NB | 0.39 | 0.44 | 0.31 | 0.44 |

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.

**CONCLUSION**

The aim for this project was to use machine learning theory to predict the rating of Yelp user 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 order to find out the model with best performance, compared different feature sets and different learning models. In this class project, includes an approach which involved the combination of topic modeling and sentiment analysis using machine learning algorithms to predict the star rating. Used different feature extraction methods such as term frequency classifier, Latent Dirichlet Allocation (LDA) and Non-negative matrix factorization (NMF) and compared and evaluated against Yelp dataset. NMF with an added sentiment layer and tf-idf model produced results which are more accurate than LDA based model. And there are still much more interesting thing to do in this project. For instance, we can keep reducing the feature dimension, using top n keywords instead of all the keywords. In Naive Bayes, we can also try different smoothing parameters.

**Future work**

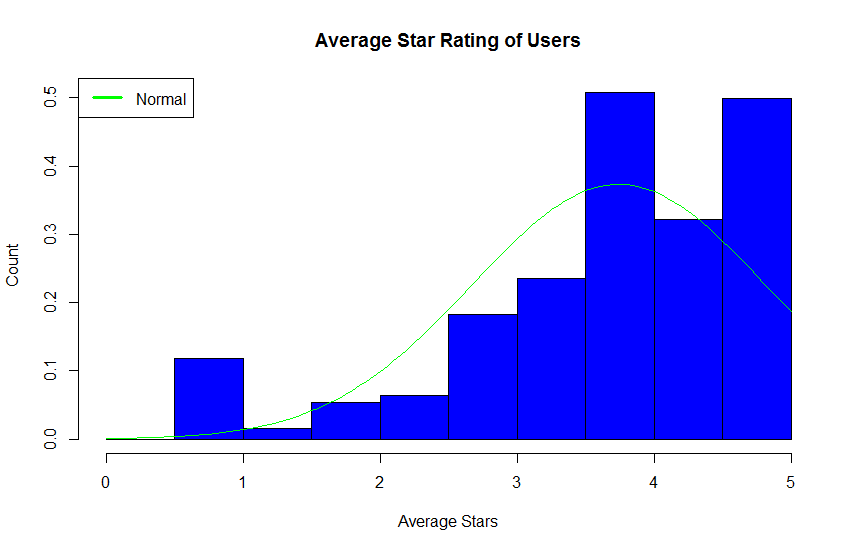
It would be interesting to observe the results when topics and sentiment extracted using such joint approach are passed as features to train the model. Furthermore, it would be interesting to test the machine learning classifiers on the features extracted by fine tuning the parameters such as choosing optimal. Finally, I would want to use libSVM with a nonlinear kernel such as Gaussian to compare with our other algorithms. Due to computational performance limitations, I was unable to implement this method.

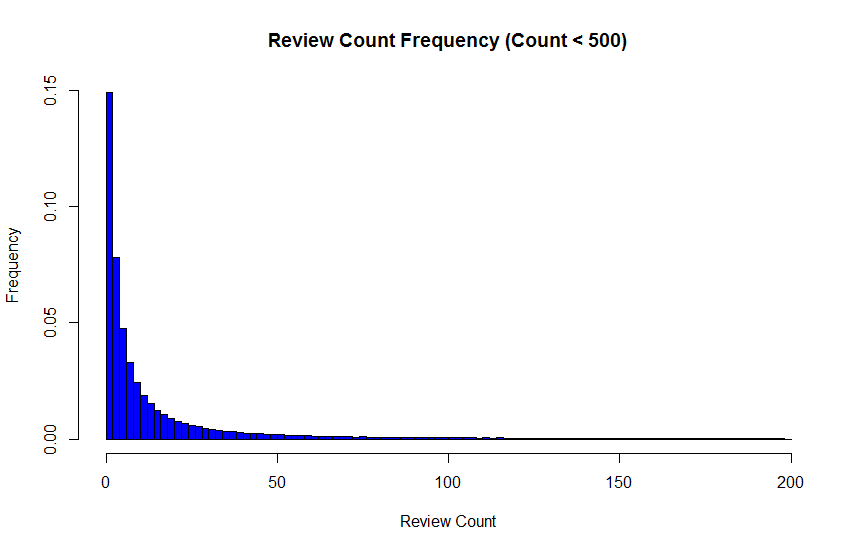
**References**

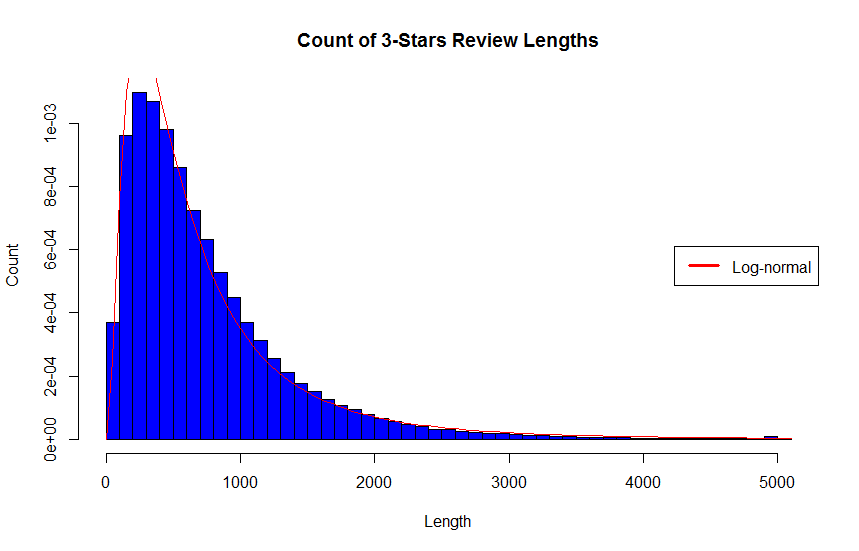
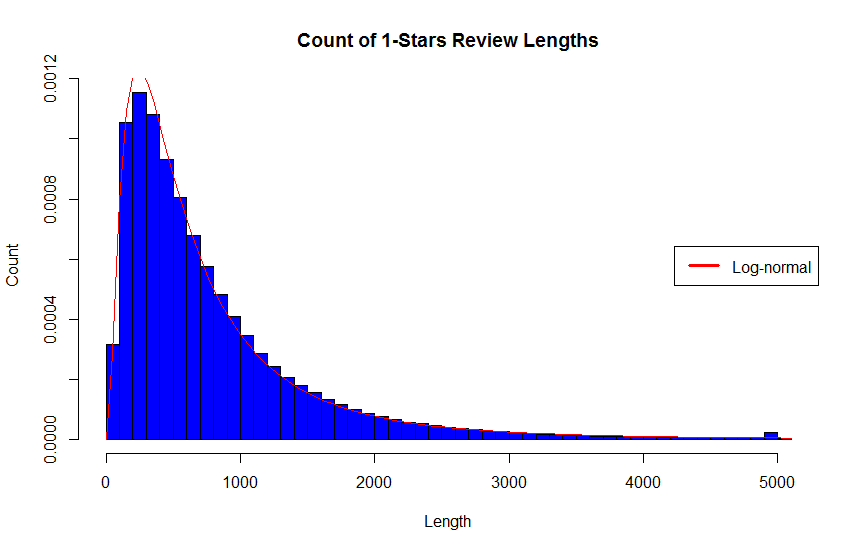
1. Scikit-Learn, <http://scikit-learn.org/stable/index.html>.
2. http://www.yelp.com/dataset challenge
3. Blei, David M., Andrew Y. Ng, and Michael I. Jordan.\Latent Dirichlet allocation." the Journal of machine learning research 3 (2003): 993-1022.
4. Lin, Chenghua, and Yulan He. \Joint sentiment/topic model for sentiment analysis." Proceedings of the 18th ACM conference on Information and knowledge management. ACM, 2009.
5. Professor Jonathan Gemmell “ETC-584 Web Data Mining for Business Intelligence” DePaul University Summer – 2016.

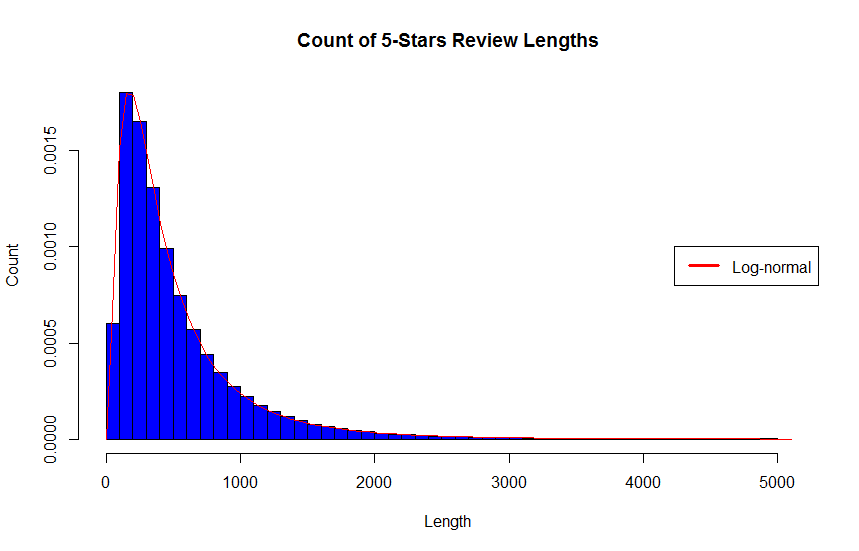
Appendix:

Plots from rstudio in the process of initial data analysis for Yelp dataset









Expected that the distribution of review lengths would differ more between the different star ratings. Intuitively, I thought 1-star and 5-star reviews would be longer because the reviewers feel strongly about one side and would write longer reviews to express their opinion. However, the data shows otherwise, and the distributions are similar distribution between the different star ratings.