Sentiment analysis on users' reviews - Yelp

Karan Desai, Kirty Vedula
Rutgers University

Abstract

Our project is on implementing sentiment analysis on user reviews obtained from Yelp's academic dataset consisting of restaurant reviews for this purpose. Yelp is a business review site that helps people find cool places, based on the informed opinions of its user community. It uses a five-point star rating system with user reviews. The main of the project is to analyze sentiments on the reviews and devise our own five-scale sentiment rating and try to match it to the star rating that the user provided. Following the n-gram approach for the analysis, we are analyzing adjective and adverbs that occur in particular sequences for this purpose. After calculating a weighted score for each review, we try to predict the users' ratings by treating multi class classification using five different classifiers and compare and contrast their results. Here, we try to determine how accurate the users are, in giving the ratings by comparing their star based rating to with our calculated results.

Reviews that are very ambiguous or belong to notorious users are obviously given a very less weight in the average calculation. Because we are able to get a good accuracy for our sentiment analysis scores it gives a much more precise calculation and a fair review system. Possible future work can include working on business IDs obtained from the dataset. The overall business ratings are no more just the average of all given ratings but rather a weighted average based on the examination of reviews by our approach and the user's accuracy ratings that we calculate.

I. Motivation

Today's businesses rely highly on how users or customers have reviewed them on various websites. Bad reviews can be a turn off to new customers. Conversely, positive reviews can grow your client list and reinforce customer loyalty. Most importantly they diverge to profit or loss. In this age of smartphones it only takes less than thirty seconds to check the review of a product before buying it or of a restaurant before dining in. This means one is highly relying on the users who review those businesses and products. We are also relying on the honesty and integrity of the user. There are a lot of factors to consider here. Some sites in this category are Zagat, Tripadvisor, Edmunds, Kbb and OpenTable.

Nielsen Study commissioned by Yelp concluded that 82% users visit the site before they intend to buy. Not only that but 44% users read the review text, 26% see the ratings, 17% see the number of users while 14% see reviews from family and friends. 85% of consumers use the internet to find local businesses. It has other material like how businesses on Yelp profit more than those who don't.

A study by Maritz research that included 3404 people found that one in four people believe the information available on ratings sites is unfair. And while the older, highly visited sites were generally perceived as more trustworthy, more than a third of visitors were still cautious of information on these sites. There may be a credibility crisis on the horizon for online review sites. If the lack of confidence in customer reviews continues, these sites could become obsolete and it is worse for consumers and businesses.

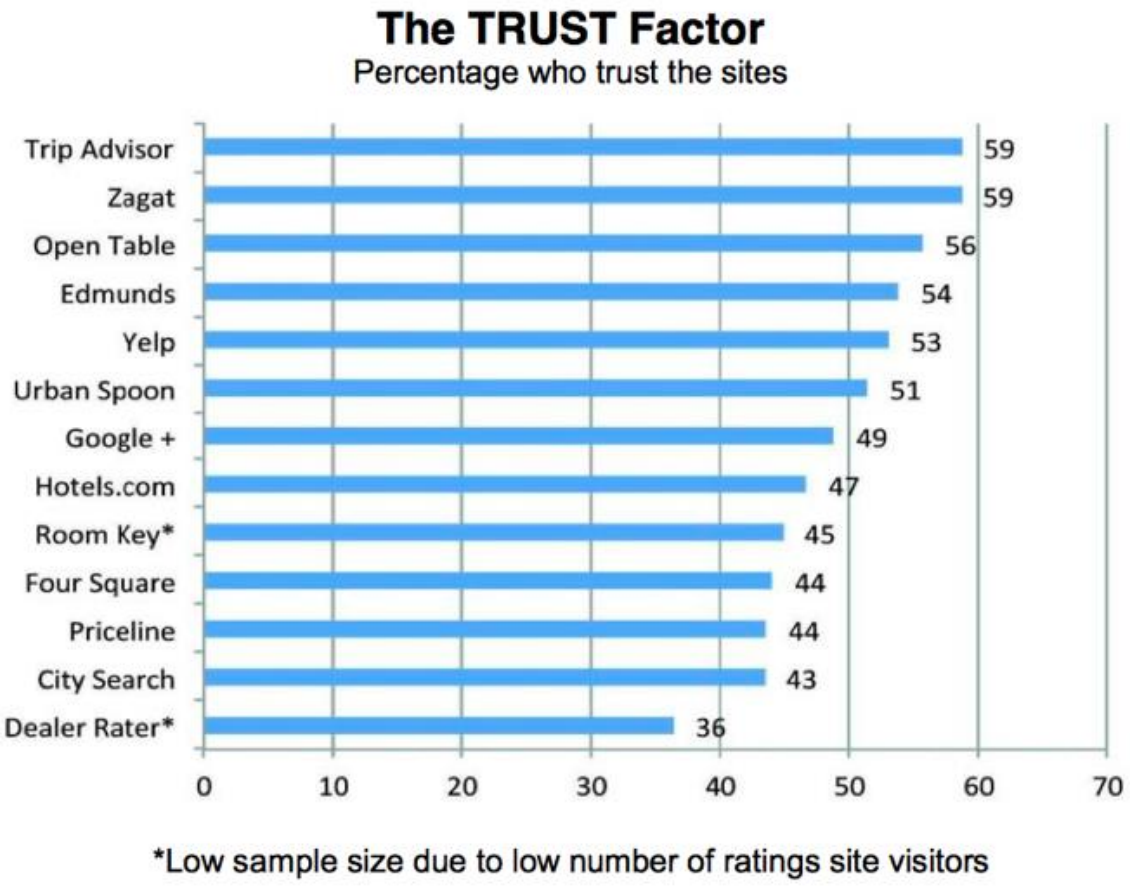


Fig 1: Maritz research about percentage of consumer who trust different websites

II. Applications of our project work

1. Generating ratings from user reviews in systems where ratings may be missing
2. Rating the user by analyzing his reviews and rating
3. Filtering certains kinds of spam based on ambiguous reviews
4. Being able to give weight to user reviews for final average calculations.

III. Problem Description

We are proposing a method that will try to eliminate such unfairness from user reviews. For the scope of this project we will work with Yelp's restaurant subset of data. First we analyze the sentiments on user reviews to predict their Likert scale ratings. Then we match these to the ones we already have to see how accurate our predictions were. We also give an accuracy rating to the users based on his how their five-star rating matches to our calculated ratings. Depending on these factors the business's average rating takes into account the individual review weights as well as weights from their users/reviewers and not simply just the average of all reviews. This comes as a part of natural language processing and artificial intelligence where the agent is developed over multiple iterations for predicting the user's ratings basing on the words given in the reviews.

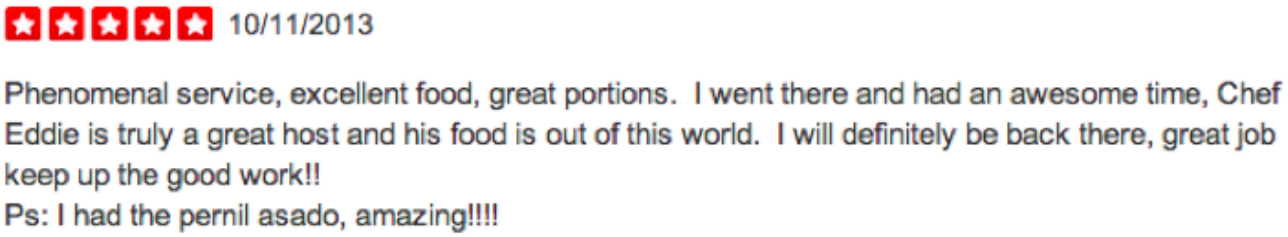


Fig 2: Yelp Sample Review with 5 Star Rating System

Unlike most analyzers which give reviews only based on positive or negative sentiment, we give a five point based rating (1 to 5) while most analyzers. We are using a bi-gram approach for sentiment analysis that is based on adjectives and adverbs occurring together or adjectives occurring by themselves. A more detailed explanation of the methodology is the next section.

Machine learning techniques are used for data analysis and decision-making tasks such as classification of categories, estimating probabilities, and data mining. However, implementing and comparing different machine learning techniques to choose the best approach can be challenging. Here, we have played with five different kind of classifiers to determine the best classifier out of all, given this situation.

IV. Methodology and implementation

In this section, we explain the flow of our project and some brief information about the background and methodology adopted. The following is the step-wise description of the project.

1. Acquire dataset
2. Split the dataset into only a user-review set
3. Use Parts-of-Speech Tagger to extract adjectives and adverbs
4. Arrange text in order to analyze bi grams. Stars-to-Words mapping only
5. Sentiment Analysis input preparation – Get
6. Analyze the text and come up with a Likert (5 point scale) rating
7. User ID's and Business ID's can be in separate locations indexed by review order

8. Obtain weighted reviews after prediction
9. Slice the data into training and testing sets
10. Build five kinds of classifiers to train the data
11. Run them on the test data
12. Perform cross validation
13. Infer from the results
14. Create visualizations for the results

Now, we describe these steps in further detail.

1. Data Set and Text Processing

On request, one can acquire a dataset from Yelp. This dataset is a specially crafted academic dataset. It has three kinds of objects in the dataset Business Objects, Review Objects and User Objects. We only need information from the User Objects for now. These objects have the review text we are looking for. There are approximately 330,000 reviews that we are going to user for this project. We will alter divide them for classification. The following is the exact JSON based data that we get in these objects.

```
DATA SET FORMAT
{
 'type': 'review',
 'business_id': (the identifier of the reviewed business),
 'user_id': (the identifier of the authoring user),
 'stars': (star rating, integer 1-5),
 'text': (review text),
 'date': (date, formatted like '2011-04-19'),
 'votes': {
          'useful': (count of useful votes),
          'funny': (count of funny votes),
          'cool': (count of cool votes)
 }
}
```

Fig 3a: Format of the dataset

```
{"votes": {"funny": 0  "useful": 0     "cool": 1} "user_id":
"kT43SxDgMGzbeXpO51f0hQ"  "review_id": "0xuZfa0t4MNWd3eIFF02ug"
"stars": 5  "date": "2009-06-09"  "text": "I'm a fan of soft
serve ice cream and Guptill's Coney Express has delicious ice
cream with many flavors.  I've tried Kurver Kreme in Colonie
Tastee Freeze in Delmar and Country Drive Inn in Clifton Park
but I think that this place has the best soft serve ice cream.
The portions are generous and the taste is very rich.  For
example the brownie sundae is decadently delicious but likely too
much for one person.  They also have cupcake sundaes which I am
looking to try soon!"  "type": "review"  "business_id":
"wbpbaWBfU54JbjLIDwERQA"}
```


Fig 3b: Raw Data Snapshot

We use Python as our main programming language. The version of python being used is 2.7.5. Most of our dataset related processing is CSV (comma separated values) format that can be read easily in any programming languages like python or using software like Microsoft excel. From this dataset we extract the fields: business_id, user_id, stars and text. These are all maintained in CSV formats.

Stars	Review Text
5	"I'm a fan of soft serve ice cream and Guptill's Coney Express has delicious ice cream with many flavors
5	"The nurses here were very attentive and wonderful. I was able to have the same surgical nurse that I
4	"Pretty great! Okay BUT I see that they do offer plenty of vegan alternatives.\n\nI was sort of skeptica
4	"The Tale of the 4-Starred leaving it looking the way it presumably did in the 70s. Rhino Records is loc
2	"As a vegan but that made the effort to put a vegetarian section on their menu to show them that it's

Fig 4. Data after extraction

Here is the flow from this step.

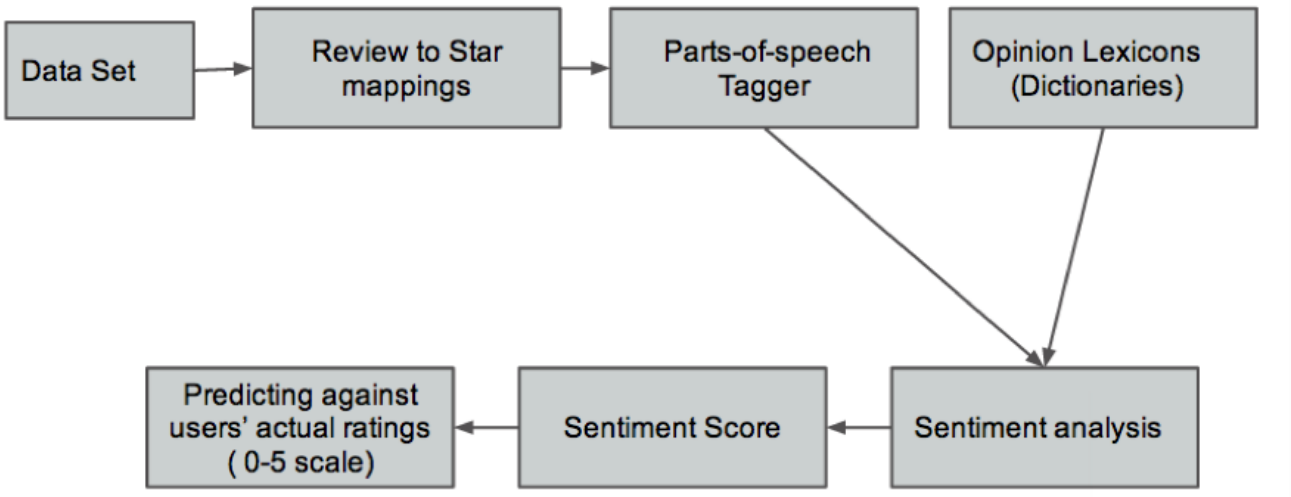


Fig 5. Flow Chart: Predicting User Ratings from Reviews

II. NLTK and POS-Tagger

The second step is using a POS-Tagger to identify the adjectives occurring in the review. The process of classifying words into their parts of speech and labeling them accordingly is known as part-of-speech tagging, POS-tagging, or simply tagging. Parts of speech are also known as word classes or lexical categories in such scopes. We also need to identify the adverbs surrounding these adjectives in the text. NLTK (Natural Language Toolkit) provides this functionality of extracting the parts of speech. We used the latest version NLTK 3.0. The following is an example from the POS-Tagger that NLTK uses.

```
POS-Tagger
>>> import nltk
>>> text=nltk.word_tokenize("This restaurant is very good .I love it here")
>>> nltk.pos_tag(text)
[('This', 'DT'), ('restaurant', 'NN'), ('is', 'VBZ'), ('very', 'RB'), ('good', 'JJ'), ('.I', 'NN'), ('love',
'NN'), ('it', 'PRP'), ('here', 'RB')]


Bi-gram
('very', 'RB'), ('good', 'JJ'),
```

Fig 6: POS-tagger processes a sequence of words, and attaches a part of speech tag to each word

Once we have an output of this kind we select words with tags 'JJ' and 'RB'. 'JJ' stands for adjective and 'RB' stands for adverbs. We are going to ignore other parts of speech for now. Now we have a csv file with user's star ratings, the words from the POS tagger and the user ratings.

III. Opinion Lexicons and Sentiment Analysis

Opinion Lexicons or dictionaries have to be obtained in order for us to do sentiment analysis. We got two text files with list of negative and positive words. Both of them have approximately 7000 words and should be enough for our scope. Having more words obviously implies more accuracy. We also have two text files that have lists of negative and positive adverbs. For the Sentiment Analysis we consider the Bi-gram approach (n-gram with n=2). What this means is that we analyze sentiments in pairs of two words wherever there is an adverb-adjective pair. This is very important step for sentiment analysis. Consider a scenario where we found the adjective "good". If the two preceding it said "not" (a negative adverb) the sentiment is inversed. On the other hand if the word preceding the adjective is "too" (a positive adverb) then it means that the positive sentiment of the adjective is boosted. So based on this logic our python program analyses the keywords and calculated a sentiment score. For each positive sentiment in the sentence we

add 1 (+1) to the total score and for each negative we subtract 1(-1). For adverbs at correct positions, positive adverbs mean multiplied by two (x 2) while negative adverb means inverting the sub score (x -1). The final sentiment likert score is obtained by first dividing the score by the number of words and then normalizing it to the 5 scale.

Star Rating	Our Rating	POS Tag	Word	POS Tag	Word
5	5	JJ	soft	JJ	delicious
5	5	RB	very	JJ	attentive
4	3.636364	RB	Pretty	JJ	great
4	4	JJ	new	JJ	aesthetic
2	5	JJ	vegetarian	JJ	worth

Fig 7. Data Snapshot after Sentiment Analysis

User_Id	Avg Deviation	Review Count
"user_id": "-iLH3Q2Wg4AMrNUXcgvliA"	0.403054771	226
"user_id": "HUmClCIluKP5Ur6X7e306Q"	0.222066287	218
"user_id": "3x8lZ-EoBhg-mw21BRITuQ"	0.087655815	184
"user_id": "itXMelaTleEjLIFWCJtnwg"	0.300926768	149
"user_id": "U4KYlRjP3KmavdPbtfOWJQ"	0.471383073	144

Fig 8. User Based deviations from actual rating

Now that we have calculated the weighted reviews, we now split the data into training and testing data sets and build classifiers. Here is the flow chart.

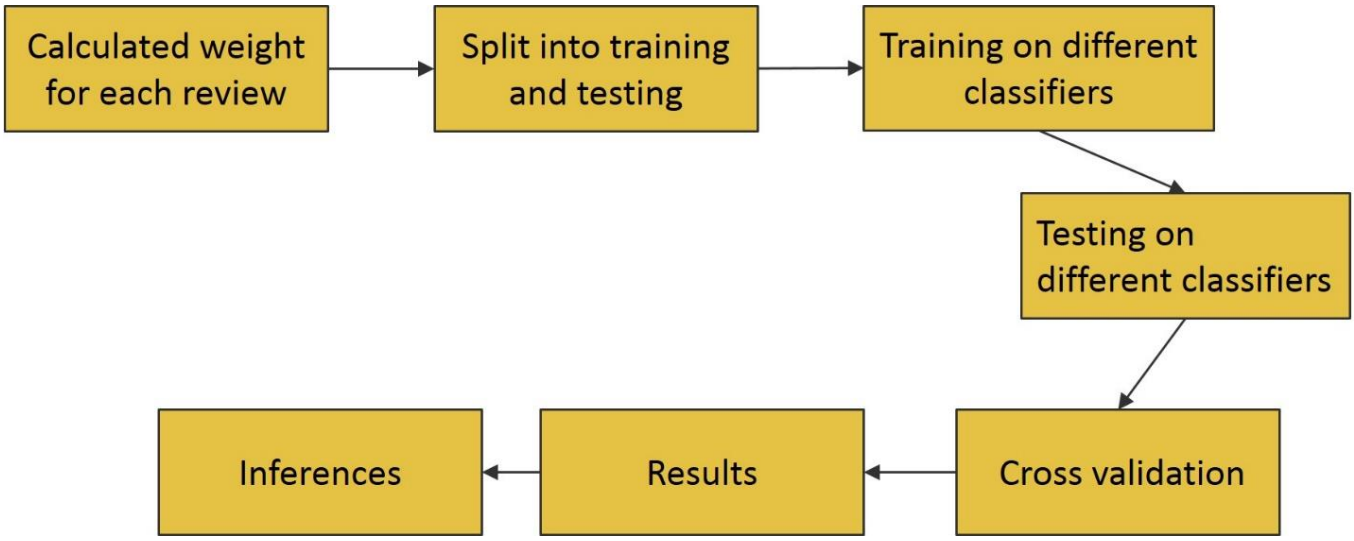


Fig 9. Flow chart for classification

IV. Multiclass Classification

Classification tasks aim to assign a predefined class to each instance. It can help to understand existing data and be used to predict how new instances. The goal of classification is to predict if the user has given a review or a comment which can match up to the client's rating. We randomly chose 70% of the documents for training and the remaining 30% for testing. Working on a 1-5 scale – randomness is spread widely (as opposed to binary models)

Here, we partition the data into training set and test set. The training set will be used to calibrate/train the model parameters. The trained model is then used to make a prediction on the test set. Predicted values will be compared with actual data to compute the confusion matrix. Confusion matrix is one way to visualize the performance of a machine learning technique.

V. Cross Validation - One-vs.-all method

Cross validation is an inherent part of machine learning. It is used to compare the performance of different predictive modeling techniques. In this example, we use holdout validation. Other techniques including k-fold and leave-one-out cross validation are also available. This method incorporates a set of binary classifiers where each one i**s** first trained to separate one class from the rest and then the multi-class classification is carried out according to the maximal output of the binary classifiers. Then, all the accuracies are combined together to give an overall accuracy in the ratings. Since the binary classifiers are obtained by training on different binary classification problems, it is unclear whether their real-valued outputs are on comparable scales situations often arise where several binary classifiers assign the same instance to their respective class (or where none does).

Classification algorithms

In this section, we summarize several supervised machine learning techniques, where one may apply one or more of the techniques and compare them to determine the most suitable ones for different datasets. We skip the mathematical details and review each approach conceptually.

1. Naïve Bayes classification

The Naive Bayes algorithm is a classification algorithm based on Bayes rule, that assumes the attributes X1,…,Xn are all conditionally independent of one another, given Y. It estimates $P(X|Y)$, the probability of features X given class Y assuming independence of the events, or that the degree of overlapping is relatively small. It is sensitive to parameter optimization

2. Logistic Regression

Logistic regression predicts the probabilities of observed outcomes for dependent variables using the values of independent variables. Since we are dealing with multinomial regression here, the outcomes have three or more possible types. Though similar to normal linear regression, logistic

regression deals with the categorical data. It acts as a discriminative classifier because we can view the distribution we are dealing with.

3. **Linear Discriminant Analysis**

We make of Fisher's linear discriminant analysis here. It is obtained from Bayes classifiers and assuming normal distribution for classes, thus making it applicable in text categorization. It expresses one dependent variable as linear combination of other features by creating an equation which will minimize the possibility of misclassifying cases into their respective groups or categories.

4. **k Nearest neighbors**

In nearest neighbor classifier, the neighbors are taken from a set of objects for which the correct classification is known. It computes the decision boundary, so that the computational complexity is a function of the boundary complexity. Categorizing query points based on their distance to points. This classifier is sensitive to the local structure of the data, and thus no training is needed explicitly. This does not require models as it computes distances to all training examples. It is susceptible to noise in the training data

5. **Support Vector Machines**

Support Vector Machines algorithm, applied to multiple classes, reduces the problem into multiple binary classification problems. Here, the binary classifier is built as a hyperplane that separates one category against the rest of the categories. For the one-versus-one approach, classification is done by a max-wins voting strategy, in which every classifier assigns the instance to one of the two classes, then the vote for the assigned class is increased by one vote, and finally the class with the most votes determines the instance classification.

Discussion

We have obtained the statistics from the Yelp for their general reviews. This matched well to the proportion of the reviews we are dealing with. The following figure describes the distribution in a pie chart.

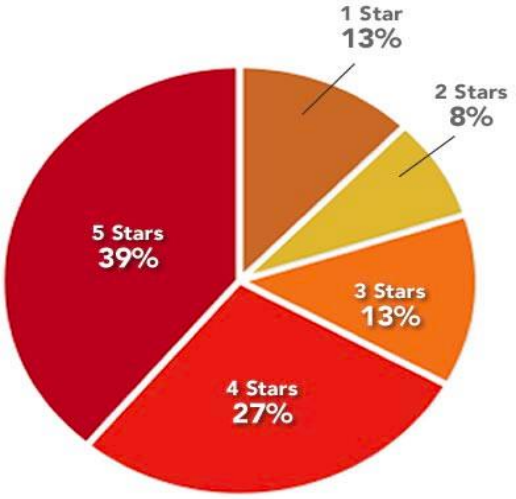


Fig 10 Yelp Rating Distribution (Complete Set of All kinds of businesses)

The following graph shows the distribution of the ratings which we have predicted.

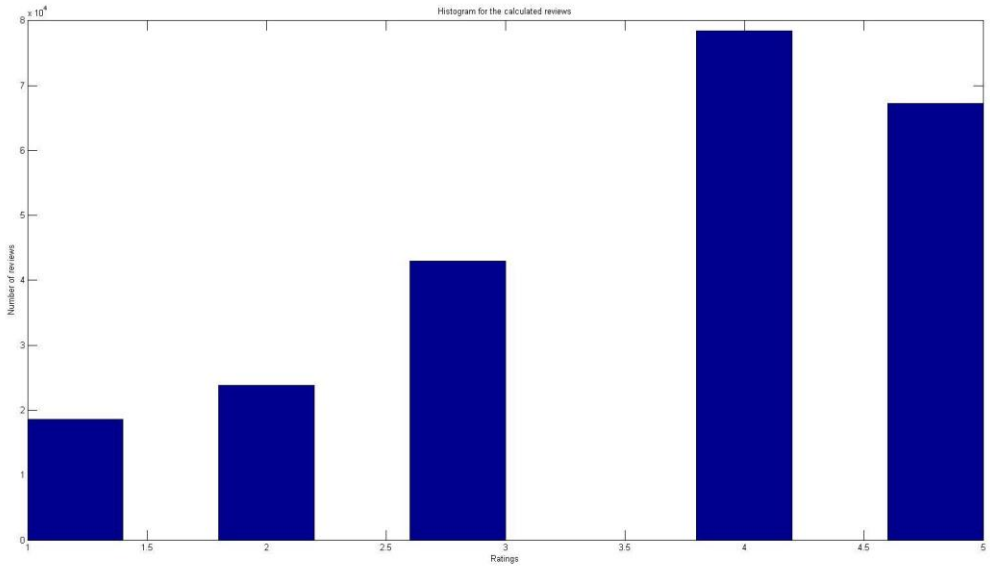
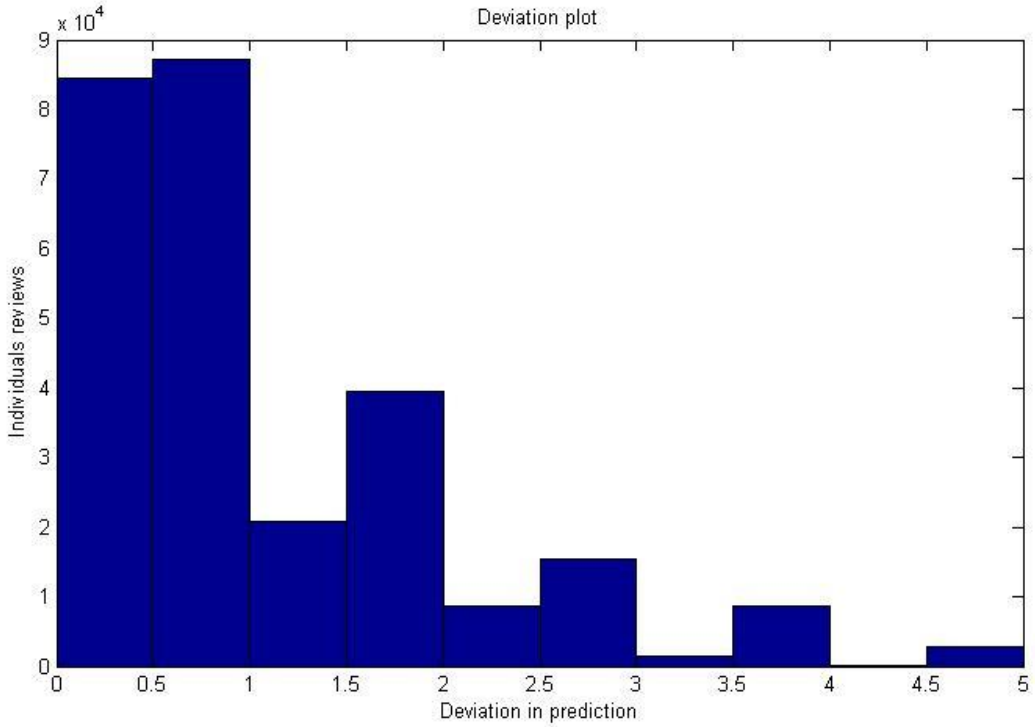


Fig 11. Distribution of the predicted ratings

The following table summarizes all the salient features of the classifiers and the accuracies achieved using each classifier.

Classification Method	Naïve Bayes Classifier	Logistic Regression	Discriminant Analysis	kNN classifier(Euclidean)	Support Vector Machines
Working	Based on estimating P(X|Y), the probability of features X given class Y	Measures the relationship among variables using prior probability scores	Expresses one dependent variable as linear combination of other features	Categorizing query points based on their distance to points	Classifies data by finding the best hyper-plane that separates all data points of one class from those of the other class
Accuracy (random = 20%)	73.67%	77.96%	73.67%	74.26%	Doesn't converge
Features	- Assumption of independence - Degree of class overlapping is small - Sensitive to parameter optimization	- Analogous to linear regression - Discriminative classifier because we can view the distribution	- Very fast compared to other classifiers - Creates an equation which will minimize the possibility of misclassifying cases into their respective groups or categories	- Does not require models - Compute distances to all training examples - Susceptible to noise in the training data - Distance measure	- Sensitive to parameter optimization - Will not operate well on non-linearly separable sets

The following plot shows the deviation in the predicted ratings.



The following graph shows the deviation from actual ratings versus the number of reviews given.

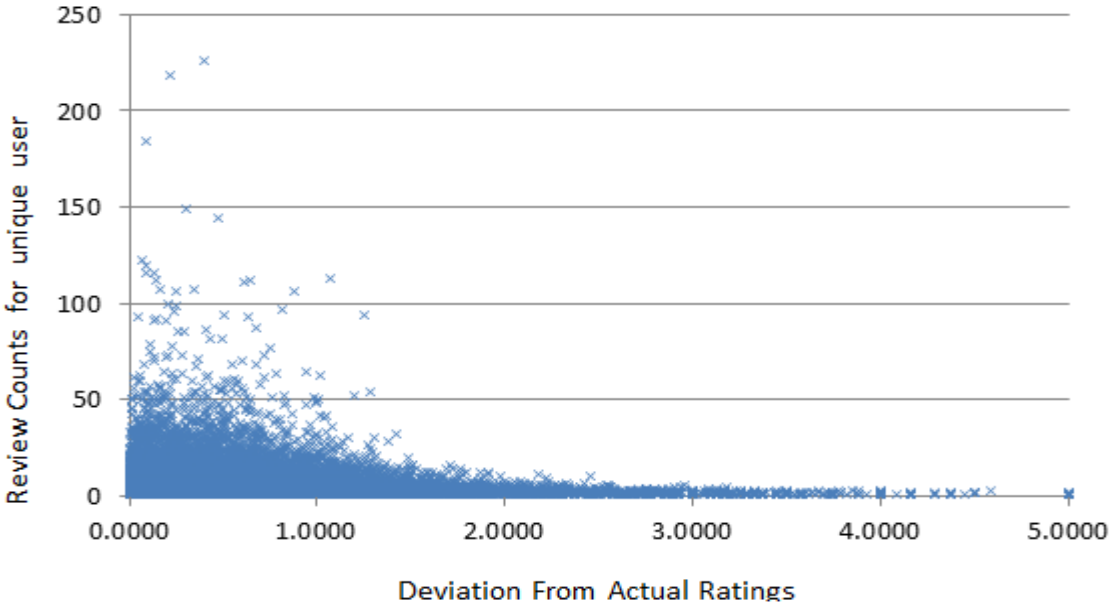


Fig. General accuracy of sentiment analysis

One classifier outperforms others in one context but fail severely in another. If one class is more likely than the others. Large number of points to capture this distinction – otherwise error. Accuracy inherently limited to the likelihood of the most likely class. For the classifier, we are solving a separate optimization problem for each class (out of 5) simultaneously. Logistic regression worked fast compared to other classifiers.

Non-convergence of SVMs

Support Vector Machines are hard classifiers, with no probability involved. They are generally good at binary classification problems. But they cannot be easily extended to their multi-class counterparts. Also as they are slow to train and complicated to implement, they fail at test instances given a different data set. They can naturally handle large dimensional data. It is sensitive to parameter optimization and not operate well on non-linearly separable sets

Future extensions

This project has the capability to be extended to be made better by including the weighted business ratings, and selecting the features in a better way. Also, if we can manage to eliminate the unnecessary features using techniques like Principal Component Analysis. Also, with better feature selection we can achieve better results. We can also include parallel processing to tackle SVMs

Individual contributions

Karan Desai contributed mainly to the first part of the project which is information retrieval and sentiment analysis. This included Data Set Text Manipulation, Parts of Speech Tagging, Gathering Opinion Lexicons, Sentiment Analysis, Generating Custom Review Rating and Calculating User's Accuracy Ratings.

Kirty Vedula has handled the second part of the project – which dealt with running classifiers. This included building the training and testing datasets, designing the five classifiers, training and testing the data sets on five different classifiers, performing cross-validation and the analysis and the inference of the results

Conclusion

To summarize the project, our method has a Likert Scale Sentiment Rating compared to binary. We have achieved a maximum accuracy of 77% using the rating of users based on their reviews. We have learnt a great deal about subtle concepts in natural language processing and general working of classifiers in this context.

Acknowledgement

This project would not have been possible without an excellent data set provided by Yelp. It has a huge potential for experiments. We would also like to thank Prof. Casimir Kulikowski, Rutgers University for his help regarding various topics on the matter.

References

1. Yelp Survey http://officialblog.yelp.com/2012/04/search-engine-lands-recent-local-consumer-review-survey-looked-at-the-way-consumer-behavior-has-changed-since-2010-intere.html

2. Maritz research http://www.maritzresearch.com/~/media/Files/MaritzResearch/Press/Maritz-Research-2013-Online-Customer-Review-Study-Release-FINAL-9-18-13.ashx

3. Opinion Lexicon - http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html

4. Stanford's NLTK - http://nltk.org/

5. Sentiment Analysis of Twitter - Data http://www.cs.columbia.edu/~julia/papers/Agarwaletal11.pdf

6. Machine learning algorithms for classification - http://www.cs.princeton.edu/~schapire/talks/picasso-minicourse.pdf