

# CS 224n Final Project: A Semantic, Supervised Classification Approach to Restaurant Reviews

**Pavani Vantimitta**

pavani@stanford.edu

Abstract

The rapid growth of E-commerce has made Customer reviews an indispensable source of information for both the potential buyer and the seller. Reviews act as a quick means of assessing the product value for the buyer and the customer feedback for the seller. Product reviews have been mined continuously for the past fifteen years or more to make them a more tractable source for users. Reviews on a product range from hundreds to thousands and make it difficult to manage an overall assessment of the product itself. This paper aims at providing an overall rating to a restaurant based on reviews collected using the semantics of the review.

## 1 Introduction

Product reviews are available in abundance on any product freely and easily accessible on the web. This does not provide a conclusive solution to a user looking to buy a product merely due to sheer quantity and the varied opinions available. Previous work in this area, has attempted various directions to address this problem. Sentiment classification has been initially studied as a cognitive linguistic problem. Work by Hearst [1] proposes a metaphoric model to determine the directionality of texts. This directionality of information is achieved by using a manually-constructed. Pang et al [2] investigates the use of several supervised machine learning methods to semantically classify movie reviews. Unsupervised learning methods have been used for semantic orientation classification like in Turney [3]. It relies on the computation of mutual information between review phrases and the words “excellent” and “poor”.

Methods other than machine learning can also be applied to classify reviews. Like Subasic and Huettner [4] use fuzzy techniques applicable to fuzzy sets to construct a lexicon and is used to analyze documents. Liu, et al [5] build linguistic affect models for six basic emotions by utilizing relationships from the Open Mind Common Sense (OMCS) knowledge base and manually specified ground truth. An affect sensing engine is then built to judge the affect of given passages. Hu and Liu [6] use the adjective synonym sets and antonym sets in WordNet [7] to judge semantic orientations of adjectives. They extend a seed set of adjectives by searching synonyms and antonyms in WordNet.

This paper aims to provide a conclusive rating based on a mix of the above related works to each restaurant that is reviewed in the data that was collected. Section 2 states the data collected and the various preprocessing methods employed. Section 3 talks about Part-of-Speech tagging used to extract semantics of the review. Section 4 explains the various ways in which a vocabulary of words was created. Section 5 elucidates the various classifiers used to classify the reviews. Section 6 is exclusively devoted to the Maximum Entropy Classifier used. Each section contains results outlining the performance.

## 2 Data Collection and Preprocessing

### 2.1 Data Set

The data set required for this exercise was to be already labeled with a rating beforehand. The initial idea was the use google base to obtain the reviews in xml files. It turned out that using google base's API (in this case Java) uses the "snippets" fees which consists of partial data. And this means that the review text is not complete. Thus this idea had to be dropped. Using the google base search (GUI) pulled up reviews based from yelp.com and thus retrieving them directly from there seemed a better choice.

The three data sets: training, validation and testing where collected from yelp.com with "Palo Alto" and "Restaurants" as search strings. The web pages retrieved from here were then crawled over using Web-Harvest [8] which is a web data extraction tool. It leverages well proved XML and text processing technologies in order to easily extract useful data from arbitrary web pages. A configuration file was written to extract the required data fields from the web pages and saved to a text file. The training set consists of 61 businesses and the validation and test set of 20 businesses each. The training set has a total of 1071 reviews, the validation set has 341 reviews and the test set has 260 reviews.

<Business> and <Reviews> are two data handles that I created to handle the data sets. Each restaurant reviewed is considered to be <Business> which consisted of one <Reviews> data structure that in-turn internally consisted of the reviews.

Business Categories hold the Category under which the cuisine of the restaurant falls under. Business Tags are some business details written by the customer or the reviewer about some of the facilities available at the restaurant (e.g. wheelchair accessibility or parking etc.) Review Opinions consists of the opinions users have left about the review, i.e. whether some prospective customer read the review and found it useful or cool or funny. These are the three opinions allowed by Yelp.com. Review Text contains the whole review comments left by the user as one whole String. Review Text words contains a list of the words that appear in each review. Review Text tags is used to fill up with the POS tags later on.

<BUSINESS>	<REVIEWS>
BUSINESS NAME	REVIEW OPINIONS
BUSINESS ADDRESS	REVIEW TEXT
BUSINESS URL	REVIEW RATING
BUSINESS CATEGORIES	REVIEW TEXT WORDS
BUSINESS TAGS	REVIEW TEXT TAGS

Table 2.1 <Business> and <Reviews> data structures with their data items.

### 2.2 Preprocessing

Some basic preprocessing methods were used to do away with some unnecessary characters that were created due to the web extraction that was done. For example, characters like "Â". Extra spaces and tabs between words were removed. Some data fields that were unnecessary were also

done away with in this stage. All punctuation marks were replaced with a space in between the words and then the punctuation marks. For example, “doing?” was replaced with “doing ?”. This was done to enable the Part-of-Speech tagger used later on to be able to successfully retrieve the correct tags without the punctuation marks causing errors.

### 3 Part-of-Speech Tagging

Stanford Log-linear Part-Of-Speech Tagger is the POS Tagger used to tag the reviews. The tagger is the main source of semantic information of the review. It was observed that the Tagger was unable to identify the end of the sentence “period” when there is no space between the last word and the period. Thus in the preprocessing step all punctuations marks were replaced with a space and then the punctuation mark. Table 3.1 shows the difference in tagging results due to this preprocessing step.

<p><b>Original Review Text</b></p>	<p>So I just got home from this place ... <b>I feel sick</b>. So lets just say I had my fair share, and my neighbor’s fair share, and my coworker’s share and so on and so on. Spicy pork tenderloin, tender skirt steak, turkey wrapped in bacon and the tasty Parmesan encrusted filet mignon. Soooo good. So good. Everything tasted awesome. Ham and pineapple; so simple yet so tasty when mixed <b>with everything else</b>. Overall the fruit in <b>general was really good</b>.</p>
<p><b>Tagged Sentence</b></p>	<p>So/RB I/PRP just/RB got/VBD home/NN from/IN this/DT place/NN .../: <b>I/PRP feel/VBP sick./VBN</b> So/NNP lets/VBZ just/RB say/VB I/PRP had/VBD my/PRP\$ fair/JJ share,/NN and/CC my/PRP\$ neighbor’s/NNS fair/JJ share,/NN and/CC my/PRP\$ coworker’s/NNS share/NN and/CC so/RB on/IN and/CC so/RB on./JJ Spicy/NNP pork/NN tenderloin,/NN tender/NN skirt/NN steak,/NN turkey/NN wrapped/VBD in/IN bacon/NN and/CC the/DT tasty/JJ Parmesan/NNP encrusted/VBD filet/NN mignon./NN Soooo/NNP good./VBZ So/RB good./JJ Everything/NN tasted/VBD awesome./RB Ham/NN and/CC pineapple;/NN so/RB simple/JJ yet/RB so/RB tasty/JJ when/WRB mixed/VBN <b>with/IN everything/NN else./VBZ</b> Overall/JJ the/DT fruit/NN in/IN <b>general/JJ was/VBD really/RB good./VBN</b></p>
<p><b>Preprocessed Review Text</b></p>	<p>So I just got home from this place ... <b>I feel sick</b> . So lets just say I had my fair share , and my neighbor's fair share , and my coworker's share and so on and so on . Spicy pork tenderloin , tender skirt steak , turkey wrapped in bacon and the tasty Parmesan encrusted filet mignon . Soooo good . So good . Everything tasted awesome . Ham and pineapple ; so simple yet so tasty when mixed <b>with everything else</b> . Overall the fruit in <b>general was really good</b> .</p>
<p><b>Preprocessed Tagged Sentence</b></p>	<p>So/RB I/PRP just/RB got/VBD home/NN from/IN this/DT place/NN .../: <b>I/PRP feel/VBP sick/JJ ./.</b> So/NNP lets/VBZ just/RB say/VB I/PRP had/VBD my/PRP\$ fair/JJ share/NN ,/, and/CC my/PRP\$ neighbor's/NNS fair/JJ share/NN ,/, and/CC my/PRP\$ coworker's/NNS share/NN and/CC so/RB on/IN and/CC so/RB on/IN ./ . Spicy/NNP pork/NN tenderloin/NN ,/, tender/NN skirt/NN steak/NN ,/, turkey/NN wrapped/VBD in/IN bacon/NN and/CC the/DT tasty/JJ Parmesan/NNP encrusted/VBD filet/NN mignon/NN ./ . Soooo/NNP good/JJ ./ . <b>So/RB</b> good/JJ ./ . Everything/NNP tasted/VBD awesome/JJ ./ . Ham/NN and/CC pineapple/NN ;/: so/RB simple/JJ yet/RB so/RB tasty/JJ when/WRB mixed/VBN <b>with/IN everything/NN else/RB ./.</b> Overall/NNP the/DT fruit/NN in/IN <b>general/JJ was/VBD really/RB good/JJ ./.</b></p>

Table 3.1: Demonstrates the change in POS tagging due preprocessing step.

In the tagged examples above, it can be noticed that the tag for each word is attached after a ‘/’ to the word itself. From the three instances highlighted in the above example, it can be concluded that the separation of a period from the last word of a sentence makes a difference in the POS tagging and thus has to be taken as a preprocessing step.

It is also necessary to notice that since no POS tagger can be perfect and that review sentences tend to follow no regular structure and are complex, some errors due to tagging cannot be avoided. The semantic information provided by POS tagging helps identify whether the tone of the review is positive or negative. The use of nouns or noun phrases indicates that a particular item is being spoken of. Similarly, an adjective indicates that that particular item is being appreciated, disliked or commented upon.

In the next section we discuss how a vocabulary is created keeping in mind the above semantics.

## **4 Vocabulary**

The semantics and orientation of some words in a review indicate more than the rest the tone of the review and the customer's opinion. Some words like "excellent", "great", "delicious", "tasty" express a positive opinion of the customer than "yuck", "cheap", "unfriendly", "mediocre", "bad", "loud" for restaurant reviews. Therefore, construction of a semantic orientation model for these type of words helps a long way in classifying a review. This approach has been used previously in [3, 6, 10]. Picking up some pointers from these works, we concentrate only on adjectives (or words tagged as adjectives) to create a vocabulary with the count of each words appearance. The rest of the section is devoted to explaining some modifications done to the vocabulary.

### **4.1 Full Vocabulary**

This is the full-fledged version of the vocabulary which pulls out all the words tagged as adjectives when the training set is POS tagged. All punctuation marks and upper case letters are replaced with spaces and lower case letters respectively. The list of businesses and their reviews were iterated through and each word tagged as an adjective was added to a HashSet to avoid duplication of words. These words were then copied into a HashMap with the count of each word's occurrence in the training set.

### **4.2 Stemmed Vocabulary**

Stemming is the process for reducing inflected (or sometimes derived) words to their stem, base or root form – generally a written word form. The stem need not be identical to the morphological root of the word; it is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root. Stemming can be a useful process to help identify that word forms in singular and plural are the same. It also helps to group words like; big, bigger and biggest are derived from the same root. Words like unimpressable and unimpressed can be considered to be from the root 'unimpress'. This helps reduction in redundant words and duplication.

Three stemmers from the WEKA project were attempted [11] namely, Lovins Stemmer, Iterated Lovins Stemmer and Snowball Stemmer. The Snowball Stemmer did not make a difference in the size of the vocabulary and thus only the Lovins and Iterated Lovins Stemmer were selected. Iterated Lovins is a iterative version of the Lovins Stemmer as can be interpreted from the name and is more severe in reducing words to their roots. This leads to grouping together words that are not from the same root and thus destroys the purpose. The Lovins stemmer is less severe in comparison.

Full Vocab	Lovins Stemmer	Iterated Lovins Stemmer
<b>expensive</b>	expens	exp
<b>authentic</b>	authent	auth
<b>fantastic</b>	fan	fan
<b>friendly</b>	frens	fr
<b>positive</b>	pos	pos

Table 4.2.1: Comparison of Stemmers used

As can be seen, this could lead to a disadvantage in confusing two different words with different words and defeats the purpose of the Stemmer. The Stemmers also have their disadvantage in overdoing words like ‘fantastic’ and ‘friendly’ which take on different forms after stemming that leave them to be mixed with a different group of meanings which could defeat our purpose of classifying reviews. Table 4.2.2 shows the vocabulary size changes due to the two different Stemmers used.

Original Vocab Size: 2729	
Stemmer	Vocab Size
<b>Lovins Stemmer</b>	2536
<b>Iterated Lovins Stemmer</b>	2423

Table 4.2.2: Vocabulary size changes due to Stemming

### 4.3 Intersecting Vocabularies

When using adjectives sometimes the words that precede the adjective also change the tone of the adjective. For example: ‘The atmosphere here is not friendly’. In this sentence the review is to be given a negative meaning but in the approach we use we only pick up the word adjective ‘friendly’ and do not consider the negative word before it. This would lead to classifying a word incorrectly. In order to avoid this, an option of removing words that appear commonly under a good and a bad review is provided. Some words could just appear once under a bad review and many times under a good review. Removing this word will be more detrimental and thus only words whose counts in each class of reviews crossing a ratio are removed. We have five classes of reviews, rating one to five, and thus rating three is considered to be neutral. Words belonging to reviews of rating 1 and 5 and words belonging to reviews of rating 2 and 4 are intersected to remove overlapping words. Table 4.3 shows the options for count ratios between which overlapping words can be removed that were attempted. Different combinations of these ratios were attempted.

Between	And
0.5	1.7
0.6	1.5
0.7	2
0.8	
1	1.5

Table 4.3: Intersection ratios for Vocabulary

### 4.4 Vocabulary Re-sizing

The last variation to the vocabulary was to remove words that had counts less than a certain value. Some words like ‘stirfried’, ‘underwhelming’, ‘parsimonious’ ‘lowprofile’ and ‘upbeat’ etc. though are relevant to restaurants are not frequently used words in a review. Thus using them in

the vocabulary is not as effective as those words that are most commonly repeated. Options for word counts less than 4, 10, 50, 100 and 500 to be removed were tried.

<b>No stemming used. Original Vocab Size: 2729</b>				
Vocab resize between	and	Resized Vocab Size	Vocab Cut off at	Final Vocab Size
>=0.6	<1.7	2648	4	521
			10	301
			50	93
			100	43
			500	3
>=0.5	<1.7	2612	4	496
			10	291
			50	92
			100	43
			500	3
>0.5	<1.7	2646	4	519
			10	299
			50	93
			100	43
			500	3
>0.7	<1.7	2659	4	532
			10	308
			50	93
			100	43
			500	3
>0.7	<1.5	2661	4	534
			10	309
			50	93
			100	43
			500	3
>0.8	<1.5	2664	4	537
			10	312
			50	93
			100	43
			500	3
>1	<1.5	2724	4	556
			10	313
			50	93
			100	43
			500	3
>1	<2	2722	4	554
			10	312
			50	93
			100	43
			500	3

Table 4.4: An example of vocabulary size changes due to the above variations

Considering that cutting off the vocabulary at count 500 leaves only 3 words, this option was dropped off. A minimal amount of three words is not sufficient to classify hundreds of reviews. Though cutting off at 50 and 100 also do not sound very convincing, at this point attempting to check the effect on the accuracy of classification was chosen.

## 5 Classifiers

### 5.1 Introduction

Classification is done into different classes of rating. There are three ways that were attempted for classification.

V1	Each rating as a different class	Five classes
V2	Rating one as a class and rating five as a different class	Two classes
V3	Rating one, two and three as a class and rating four and five as a class	Two classes

Table 5.1.1: Variations of classification

The first method of classification was not successful unanimously with regards to any classifier that was used. One obvious reason for this was that as is the case in the real world reviews, rating one and two (interpreted as bad reviews) reviews are a rare occurrence and thus with even 1971 reviews in total for training and around 250 reviews for both validation and testing the classification results are abysmal. Rating three, four and five do a little better but not very promising.

The second variation does not work well for the same reasons as above. Rating one reviews are infinitesimally small and thus the performance on the side of rating five is very high. It can almost be interpreted as if all reviews are directly classified as rating five and those that are misclassified are just to be interpreted as errors.

The third variation seems to be the best strategy to follow by classifying reviews of rating one, two and three as negative reviews and classifying reviews of rating four and five as positive reviews.

A total of eleven different classifiers were attempted on the classification task. All the classifiers are picked up from the WEKA Project [11]. All these classifiers provide for instance weights to be given for each training instance added or test instance. After a rigorous test of all the possible weights to each of the possible classes in the three variations discussed above, by incrementing the weights in step sizes of 0.1 until 1, it was concluded that assigning equal weights gave the best performance. Though giving more weights to rating one in Variations V1 and V2 than rating five should overcome the problem of lesser training examples, it still does not do justice to the precision, recall and f-score values of the class rating one belongs to. Thus equal weights seemed the best option. This testing was done for the Naïve Bayes classifier.

Classifiers	
Naïve Bayes – Baseline Classifier	
Bayes Net	K-star
Logistic Regression	LWL
Classification via Regression	IB-1
Support Vector Machine	IB-5
Classification via Clustering	IB-10

Table 5.1.2: Various classifiers used for classification

## 5.2 Naïve Bayes

Naïve Bayes classifier is employed here as the Baseline classifier for this task.

### 5.2.1 The first set of results is for classification of Variation 3:

We can hypothesize here that as the vocabulary cut off increases the accuracy and the accuracy and thereby the other measures should also increase. The resizing of the vocabulary should also show the same pattern to an extent. Note: The red highlight in the tables indicates the best values for that measure.

No Stemming involved									
Vocab resize between		Vocab cut off at	Accuracy (%)	Precision of class one	Recall of class one	F-score of class one	Precision of class two	Recall of class two	F-score of class two
0.5	1.5	4	73.8462	0.344	0.429	0.382	0.859	0.81	0.834
0.5	1.5	10	73.8462	0.344	0.429	0.382	0.859	0.81	0.834
0.5	1.7	4	73.8462	0.344	0.429	0.382	0.859	0.81	0.834
0.5	1.7	10	73.8462	0.344	0.429	0.382	0.859	0.81	0.834
0.5	2	4	73.8462	0.344	0.429	0.382	0.859	0.81	0.834
0.5	2	10	73.8462	0.344	0.429	0.382	0.859	0.81	0.834
0.5	1.5	0	74.2308	0.35	0.429	0.385	0.86	0.815	0.837
0.5	1.7	0	74.2308	0.35	0.429	0.385	0.86	0.815	0.837
0.5	2	0	74.2308	0.35	0.429	0.385	0.86	0.815	0.837
0.6	1.5	10	74.2308	0.35	0.429	0.385	0.86	0.815	0.837
0.6	1.7	10	74.2308	0.35	0.429	0.385	0.86	0.815	0.837
0.6	2	10	74.2308	0.35	0.429	0.385	0.86	0.815	0.837
0.7	1.5	0	74.2308	0.35	0.429	0.385	0.86	0.815	0.837
0.7	1.5	4	74.2308	0.35	0.429	0.385	0.86	0.815	0.837
0.7	1.5	10	74.2308	0.35	0.429	0.385	0.86	0.815	0.837
0.7	1.7	0	74.2308	0.35	0.429	0.385	0.86	0.815	0.837
0.7	1.7	4	74.2308	0.35	0.429	0.385	0.86	0.815	0.837
0.7	1.7	10	74.2308	0.35	0.429	0.385	0.86	0.815	0.837
0.7	2	0	74.2308	0.35	0.429	0.385	0.86	0.815	0.837
0.7	2	4	74.2308	0.35	0.429	0.385	0.86	0.815	0.837
0.7	2	10	74.2308	0.35	0.429	0.385	0.86	0.815	0.837
0.8	1.5	0	74.2308	0.35	0.429	0.385	0.86	0.815	0.837
0.8	1.5	4	74.2308	0.35	0.429	0.385	0.86	0.815	0.837
0.8	1.5	10	74.2308	0.35	0.429	0.385	0.86	0.815	0.837
0.8	1.7	0	74.2308	0.35	0.429	0.385	0.86	0.815	0.837

0.8	1.7	4	74.2308	0.35	0.429	0.385	0.86	0.815	0.837
0.8	1.7	10	74.2308	0.35	0.429	0.385	0.86	0.815	0.837
0.8	2	0	74.2308	0.35	0.429	0.385	0.86	0.815	0.837
0.8	2	4	74.2308	0.35	0.429	0.385	0.86	0.815	0.837
0.8	2	10	74.2308	0.35	0.429	0.385	0.86	0.815	0.837
0.6	1.5	0	74.6154	0.356	0.429	0.389	0.861	0.82	0.84
0.6	1.5	4	74.6154	0.356	0.429	0.389	0.861	0.82	0.84
0.6	1.7	0	74.6154	0.356	0.429	0.389	0.861	0.82	0.84
0.6	1.7	4	74.6154	0.356	0.429	0.389	0.861	0.82	0.84
0.6	2	0	74.6154	0.356	0.429	0.389	0.861	0.82	0.84
0.6	2	4	74.6154	0.356	0.429	0.389	0.861	0.82	0.84
0.5	1.5	50	75.3846	0.358	0.388	0.373	0.855	0.839	0.847
0.5	1.7	50	75.3846	0.358	0.388	0.373	0.855	0.839	0.847
0.5	2	50	75.3846	0.358	0.388	0.373	0.855	0.839	0.847
0.6	1.5	50	75.3846	0.358	0.388	0.373	0.855	0.839	0.847
0.6	1.7	50	75.3846	0.358	0.388	0.373	0.855	0.839	0.847
0.6	2	50	75.3846	0.358	0.388	0.373	0.855	0.839	0.847
0.7	1.5	50	75.3846	0.358	0.388	0.373	0.855	0.839	0.847
0.7	1.7	50	75.3846	0.358	0.388	0.373	0.855	0.839	0.847
0.7	2	50	75.3846	0.358	0.388	0.373	0.855	0.839	0.847
0.8	1.5	50	75.3846	0.358	0.388	0.373	0.855	0.839	0.847
0.8	1.7	50	75.3846	0.358	0.388	0.373	0.855	0.839	0.847
0.8	2	50	75.3846	0.358	0.388	0.373	0.855	0.839	0.847
1	1.5	50	75.3846	0.358	0.388	0.373	0.855	0.839	0.847
1	1.7	50	75.3846	0.358	0.388	0.373	0.855	0.839	0.847
1	2	50	75.3846	0.358	0.388	0.373	0.855	0.839	0.847
1	1.5	0	75.7692	0.379	0.449	0.411	0.866	0.829	0.847
1	1.5	4	75.7692	0.379	0.449	0.411	0.866	0.829	0.847
1	1.5	10	75.7692	0.379	0.449	0.411	0.866	0.829	0.847
1	1.7	0	75.7692	0.379	0.449	0.411	0.866	0.829	0.847
1	1.7	4	75.7692	0.379	0.449	0.411	0.866	0.829	0.847
1	1.7	10	75.7692	0.379	0.449	0.411	0.866	0.829	0.847
1	2	0	75.7692	0.379	0.449	0.411	0.866	0.829	0.847
1	2	4	75.7692	0.379	0.449	0.411	0.866	0.829	0.847
1	2	10	75.7692	0.379	0.449	0.411	0.866	0.829	0.847
0.5	1.5	100	78.8462	0.429	0.367	0.396	0.858	0.886	0.872
0.5	1.7	100	78.8462	0.429	0.367	0.396	0.858	0.886	0.872
0.5	2	100	78.8462	0.429	0.367	0.396	0.858	0.886	0.872
0.6	1.5	100	78.8462	0.429	0.367	0.396	0.858	0.886	0.872
0.6	1.7	100	78.8462	0.429	0.367	0.396	0.858	0.886	0.872
0.6	2	100	78.8462	0.429	0.367	0.396	0.858	0.886	0.872
0.7	1.5	100	78.8462	0.429	0.367	0.396	0.858	0.886	0.872
0.7	1.7	100	78.8462	0.429	0.367	0.396	0.858	0.886	0.872
0.7	2	100	78.8462	0.429	0.367	0.396	0.858	0.886	0.872
0.8	1.5	100	78.8462	0.429	0.367	0.396	0.858	0.886	0.872
0.8	1.7	100	78.8462	0.429	0.367	0.396	0.858	0.886	0.872
0.8	2	100	78.8462	0.429	0.367	0.396	0.858	0.886	0.872
1	1.5	100	78.8462	0.429	0.367	0.396	0.858	0.886	0.872
1	1.7	100	78.8462	0.429	0.367	0.396	0.858	0.886	0.872
1	2	100	78.8462	0.429	0.367	0.396	0.858	0.886	0.872

Table 5.2.1.1: Shows the Summary of the results for Naïve Bayes Classifier without Stemming.

Table 5.2.1 shows that the vocabulary resizing does not have as profound an effect as was expected. For no stemming, the accuracy decreases by half a point when resizing starts from ratio 0.5, 0.7 and 0.8 for no vocabulary count cut off or vocabulary cutoff at 4 and 10. For no

stemming, the accuracy increases by a point when resizing starts from ratio one for no vocabulary count cut off or vocabulary cutoff at 4 and 10. For vocabulary cut offs at 50 and 100 the accuracy and other measures remain the same. Similarly for Iterated Lovins Stemmer, the accuracy drops by half a point for all vocabulary cut off values except 100 starting from ration 0.7 for vocabulary resizing. Lovins Stemmer follows the same format like Iterated Lovins Stemmer.

Table 5.2.2 shows that using a Stemmer is better with regards to Accuracy and Recall for class two but reduced the precision of class two and recall, F-score for class one.

Stemming	Vocab Cut off at	Accuracy (%)	Precision of class one	Recall of class one	F-score of class one	Precision of class two	Recall of class two	F-score of class two
None	100	78.8462	0.429	0.367	0.396	0.858	0.886	0.872
Lovins Stemmer	100	79.2308	0.429	0.306	0.357	0.849	0.905	0.876
Iterated Lovins Stemmer	100	79.2308	0.429	0.306	0.357	0.849	0.905	0.876

Table 5.2.1.2: Shows the best results for different Stemmers.

## 5.2.2 Classification of type Variation 2:

Unlike in the above case, the precision, recall and F-score suffer badly for class one for all the different stemmers used. These values are very close to zero. The accuracy levels though are really high in the 90's but since the precision and recall levels are really low on the other class, the reason for this is as hypothesized in the starting of this section. We could almost consider this classification as blindly labeling all the reviews as class two and the ones that are actually class one are labeled wrong, but since their number is smaller in number, the accuracy is still high. Another pattern found here is that increasing the vocabulary cutoff to 50 or 100 actually decreases the accuracy but is still favorable because it increase the precision, recall and F-score for class one. This is justifiable because more than the overall accuracy we are more concerned about equal performance on both the classes, which is judged by the precision, recall and other measures.

Stemming	Vocab Cut off at	Accuracy (%)	Precision of class one	Recall of class one	F-score of class one	Precision of class two	Recall of class two	F-score of class two
None	0/ 4	94.231	0	0	0	0.98	0.961	0.97
Lovins Stemmer	100	97.308	0	0	0	0.981	0.992	0.986
Iterated Lovins Stemmer	100	97.308	0	0	0	0.981	0.992	0.986

Table 5.2.2.1: Shows the best results for all Stemmers

The accuracy follows the same pattern as in Section 5.2.1 by increasing with the usage of Stemmers. But the use of Stemmers in damaging to the measure of class one, since all the trials with both the Stemmers produced zero for all measures for class one. Using the Stemmers gives a higher accuracy to vocabulary cutoff at 100. There is a steady decrease in accuracy from vocabulary cutoff from zero to 50 and then a sudden increase for cutoff at 100 in the case of Iterated Lovins Stemmer. There is a constant accuracy level between vocabulary cutoffs at zero, 4 and 10 except for a sudden decrease at cutoff zero between vocabulary resizing ratios 0.7 to 1. The accuracy decreases for cutoff at 50 and then increases by 2 points for cutoff at 100. As can clearly be seen there is no reasonable explanation to this except to assume that the classifier classifies every

review as class two blindly. This hypothesis proved to be right when the confusion matrices were observed.

### 5.2.3 Classification for Variation 1:

Stemming	Vocab Cut off at	Accuracy (%)	Precision of class one	Recall of class one	F-score of class one	Precision of class two	Recall of class two	F-score of class two
None	100	41.1538	<0.5	<0.5	<0.5	<0.5	<0.5	<0.5
Lovins Stemmer	100/10/4/0	43.0769	<0.5	<0.5	<0.5	<0.5	<0.5	<0.5
Iterated Lovins Stemmer	100	43.0769	<0.5	<0.5	<0.5	<0.5	<0.5	<0.5

Table 5.2.3.1: Shows the best results for all Stemmers

The results for this variation are all below 50% accuracy. The precision, recall and F-score for all classes are below 0.5 and for class one most of the times it is close to zero. Classes two and three have values closer to 0.25. Thus on the whole the accuracy increases because of the Stemmers but this variation is not very promising. Moving ahead from here, we will consider only variation 3 for classification.

### 5.3 Logistic Regression

This classifier is again a part of the WEKA project and they modify the Logistic Regression algorithm to suit the purposes of classification as explained below.

If there are k classes for n instances with m attributes, the parameter matrix B to be calculated will be an m\*(k-1) matrix. The probability for class j with the exception of the last class is

$$P_j(X_i) = \exp(X_i B_j) / ((\sum_{j=1..(k-1)} \exp(X_i * B_j)) + 1)$$

The last class has probability

$$1 - (\sum_{j=1..(k-1)} P_j(X_i)) = 1 / ((\sum_{j=1..(k-1)} \exp(X_i * B_j)) + 1)$$

The (negative) multinomial log-likelihood is thus:

$$L = -\sum_{i=1..n} \{ \sum_{j=1..(k-1)} (Y_{ij} * \ln(P_j(X_i))) + (1 - (\sum_{j=1..(k-1)} Y_{ij})) * \ln(1 - \sum_{j=1..(k-1)} P_j(X_i)) \} + \text{ridge} * (B^2)$$

In order to find the matrix B for which L is minimised, a Quasi-Newton Method is used to search for the optimized values of the m\*(k-1) variables. Although original Logistic Regression does not deal with instance weights, they modify the algorithm a little bit to handle the instance weights.

No Stemming involved									
Vocab resize between		Vocab cut off at	Accuracy (%)	Precision of class one	Recall of class one	F-score of class one	Precision of class two	Recall of class two	F-score of class two
0.5	1.5	0	72.692	0.238	0.204	0.22	0.821	0.848	0.834
0.5	1.5	4	75	0.265	0.184	0.217	0.823	0.882	0.851
0.5	1.5	10	76.154	0.367	0.367	0.367	0.853	0.853	0.853
0.5	1.5	50	78.846	0.35	0.143	0.203	0.825	0.938	0.878
0.5	1.5	100	80	0.385	0.102	0.161	0.822	0.962	0.886
0.5	1.7	0	71.154	0.205	0.184	0.194	0.815	0.834	0.824
0.5	1.7	4	73.846	0.243	0.184	0.209	0.821	0.867	0.843
0.5	1.7	10	76.154	0.356	0.327	0.34	0.847	0.863	0.854
0.5	1.7	50	78.846	0.35	0.143	0.203	0.825	0.938	0.878
0.5	1.7	100	80	0.385	0.102	0.161	0.822	0.962	0.886
0.5	2	0	71.154	0.205	0.184	0.194	0.815	0.834	0.824
0.5	2	4	73.846	0.243	0.184	0.209	0.821	0.867	0.843
0.5	2	10	76.154	0.356	0.327	0.34	0.847	0.863	0.854
0.5	2	50	78.846	0.35	0.143	0.203	0.825	0.938	0.878
0.5	2	100	80	0.385	0.102	0.161	0.822	0.962	0.886
0.6	1.5	0	72.692	0.238	0.204	0.22	0.821	0.848	0.834
0.6	1.5	4	75	0.265	0.184	0.217	0.823	0.882	0.851
0.6	1.5	10	76.539	0.375	0.367	0.371	0.854	0.858	0.856
0.6	1.5	50	78.846	0.35	0.143	0.203	0.825	0.938	0.878
0.6	1.5	100	80	0.385	0.102	0.161	0.822	0.962	0.886
0.6	1.7	0	71.154	0.205	0.184	0.194	0.815	0.834	0.824
0.6	1.7	4	75.385	0.286	0.204	0.238	0.827	0.882	0.853
0.6	1.7	10	76.923	0.378	0.347	0.362	0.851	0.867	0.859
0.6	1.7	50	78.846	0.35	0.143	0.203	0.825	0.938	0.878
0.6	1.7	100	80	0.385	0.102	0.161	0.822	0.962	0.886
0.6	2	0	71.154	0.205	0.184	0.194	0.815	0.834	0.824
0.6	2	4	75.385	0.286	0.204	0.238	0.827	0.882	0.853
0.6	2	10	76.923	0.378	0.347	0.362	0.851	0.867	0.859
0.6	2	50	78.846	0.35	0.143	0.203	0.825	0.938	0.878
0.6	2	100	80	0.385	0.102	0.161	0.822	0.962	0.886
0.7	1.5	0	72.692	0.238	0.204	0.22	0.821	0.848	0.834
0.7	1.5	4	77.308	0.344	0.224	0.272	0.833	0.9	0.866
0.7	1.5	10	78.077	0.382	0.265	0.313	0.841	0.9	0.87
0.7	1.5	50	78.846	0.35	0.143	0.203	0.825	0.938	0.878
0.7	1.5	100	80	0.385	0.102	0.161	0.822	0.962	0.886
0.7	1.7	0	69.615	0.222	0.245	0.233	0.82	0.801	0.811
0.7	1.7	4	74.615	0.27	0.204	0.233	0.825	0.872	0.848
0.7	1.7	10	77.308	0.344	0.224	0.272	0.833	0.9	0.866
0.7	1.7	50	78.846	0.35	0.143	0.203	0.825	0.938	0.878
0.7	1.7	100	80	0.385	0.102	0.161	0.822	0.962	0.886
0.7	2	0	69.615	0.222	0.245	0.233	0.82	0.801	0.811
0.7	2	4	74.615	0.27	0.204	0.233	0.825	0.872	0.848
0.7	2	10	77.308	0.344	0.224	0.272	0.833	0.9	0.866
0.7	2	50	78.846	0.35	0.143	0.203	0.825	0.938	0.878
0.7	2	100	80	0.385	0.102	0.161	0.822	0.962	0.886
0.8	1.5	0	73.846	0.289	0.265	0.277	0.833	0.848	0.84
0.8	1.5	4	77.308	0.344	0.224	0.272	0.833	0.9	0.866
0.8	1.5	10	78.846	0.35	0.143	0.203	0.825	0.938	0.878
0.8	1.5	50	78.846	0.35	0.143	0.203	0.825	0.938	0.878
0.8	1.5	100	80	0.385	0.102	0.161	0.822	0.962	0.886
0.8	1.7	0	71.923	0.26	0.265	0.263	0.829	0.825	0.827
0.8	1.7	4	75.385	0.286	0.204	0.238	0.827	0.882	0.853
0.8	1.7	50	78.846	0.35	0.143	0.203	0.825	0.938	0.878

0.8	1.7	10	79.231	0.419	0.265	0.325	0.843	0.915	0.877
0.8	1.7	100	80	0.385	0.102	0.161	0.822	0.962	0.886
0.8	2	0	71.923	0.26	0.265	0.263	0.829	0.825	0.827
0.8	2	4	75.385	0.286	0.204	0.238	0.827	0.882	0.853
0.8	2	50	78.846	0.35	0.143	0.203	0.825	0.938	0.878
0.8	2	10	79.231	0.419	0.265	0.325	0.843	0.915	0.877
0.8	2	100	80	0.385	0.102	0.161	0.822	0.962	0.886
1	1.5	0	73.846	0.289	0.265	0.277	0.833	0.848	0.84
1	1.5	50	78.846	0.35	0.143	0.203	0.825	0.938	0.878
1	1.5	10	79.615	0.441	0.306	0.361	0.85	0.91	0.879
1	1.5	4	80	0.385	0.102	0.161	0.822	0.962	0.886
1	1.5	100	80	0.385	0.102	0.161	0.822	0.962	0.886
1	1.7	0	73.462	0.308	0.327	0.317	0.841	0.829	0.835
1	1.7	4	78.462	0.387	0.245	0.3	0.838	0.91	0.873
1	1.7	50	78.846	0.35	0.143	0.203	0.825	0.938	0.878
1	1.7	100	80	0.385	0.102	0.161	0.822	0.962	0.886
1	1.7	10	80.385	0.469	0.306	0.37	0.851	0.919	0.884
1	2	0	73.462	0.308	0.327	0.317	0.841	0.829	0.835
1	2	4	78.462	0.387	0.245	0.3	0.838	0.91	0.873
1	2	50	78.846	0.35	0.143	0.203	0.825	0.938	0.878
1	2	100	80	0.385	0.102	0.161	0.822	0.962	0.886
1	2	10	80.385	0.469	0.306	0.37	0.851	0.919	0.884

Table 5.3.1: Shows the measures for Logistic regression Classifier

There is no set pattern that the results follow contrary to our hypotheses. The vocabulary cutoff values do not encourage any pattern except that within a vocabulary resize ratio limits the accuracy steadily increases with increasing vocabulary cutoff values. But when compared to each level of ratios, there is again no pattern.

Stemming	Vocab resize Ratios	Vocab Cut off at	Accuracy (%)	Precision of class one	Recall of class one	F-score of class one	Precision of class two	Recall of class two	F-score of class two
None	1 and 1.7/2	10	80.385	0.469	0.306	0.37	0.851	0.919	0.884
Lovins Stemmer	All	50	81.923	0.563	0.184	0.277	0.836	0.967	0.897
Iterated Lovins Stemmer	All	50	81.923	0.563	0.184	0.277	0.836	0.967	0.897

Table 5.3.2: Shows the results for different Stemmers.

As expected the accuracy increases by using the Stemmer but the recall, f-score of class one decreases and the precision of class two. Just as in the Naïve Bayes results, there cannot be a conclusion drawn on whether using a Stemmer is of any consequence or not. But we can also notice that the two stemmers do not produce different results. Though in Section 4 we did notice that the Iterated Lovins Stemmer was more drastic than the Lovins stemmer at reducing a word to its word form but yet, the results come out to be the same in all cases. From here onwards we will consider only the Lovins Stemmer results.

## 5.4 Classification via Clustering

No Stemming involved									
Vocab resize between		Vocab cut off at	Accuracy (%)	Precision of class one	Recall of class one	F-score of class one	Precision of class two	Recall of class two	F-score of class two
0.5	1.5	100	58.462	0.174	0.1	0.127	0.804	0.886	0.843
0.5	1.7	100	58.462	0.174	0.1	0.127	0.804	0.886	0.843
0.5	2	100	58.462	0.174	0.1	0.127	0.804	0.886	0.843
0.6	1.5	100	58.462	0.174	0.1	0.127	0.804	0.886	0.843
0.6	1.7	100	58.462	0.174	0.1	0.127	0.804	0.886	0.843
0.6	2	100	58.462	0.174	0.1	0.127	0.804	0.886	0.843
0.7	1.5	100	58.462	0.174	0.1	0.127	0.804	0.886	0.843
0.7	1.7	100	58.462	0.174	0.1	0.127	0.804	0.886	0.843
0.7	2	100	58.462	0.174	0.1	0.127	0.804	0.886	0.843
0.8	1.5	100	58.462	0.174	0.1	0.127	0.804	0.886	0.843
0.8	1.7	100	58.462	0.174	0.1	0.127	0.804	0.886	0.843
0.8	2	100	58.462	0.174	0.1	0.127	0.804	0.886	0.843
1	1.5	100	58.462	0.174	0.1	0.127	0.804	0.886	0.843
1	1.7	100	58.462	0.174	0.1	0.127	0.804	0.886	0.843
1	2	100	58.462	0.174	0.1	0.127	0.804	0.886	0.843
0.5	1.5	50	67.308	0.111	0.043	0.063	0.797	0.915	0.852
0.5	1.7	50	67.308	0.111	0.043	0.063	0.797	0.915	0.852
0.5	2	50	67.308	0.111	0.043	0.063	0.797	0.915	0.852
0.6	1.5	50	67.308	0.111	0.043	0.063	0.797	0.915	0.852
0.6	1.7	50	67.308	0.111	0.043	0.063	0.797	0.915	0.852
0.6	2	50	67.308	0.111	0.043	0.063	0.797	0.915	0.852
0.7	1.5	50	67.308	0.111	0.043	0.063	0.797	0.915	0.852
0.7	1.7	50	67.308	0.111	0.043	0.063	0.797	0.915	0.852
0.7	2	50	67.308	0.111	0.043	0.063	0.797	0.915	0.852
0.8	1.5	50	67.308	0.111	0.043	0.063	0.797	0.915	0.852
0.8	1.7	50	67.308	0.111	0.043	0.063	0.797	0.915	0.852
0.8	2	50	67.308	0.111	0.043	0.063	0.797	0.915	0.852
1	1.5	50	67.308	0.111	0.043	0.063	0.797	0.915	0.852
1	1.7	50	67.308	0.111	0.043	0.063	0.797	0.915	0.852
1	2	50	67.308	0.111	0.043	0.063	0.797	0.915	0.852
0.5	1.5	10	75	0.059	0.021	0.031	0.808	0.924	0.862
0.5	1.7	10	75	0.059	0.021	0.031	0.808	0.924	0.862
0.5	2	10	75	0.059	0.021	0.031	0.808	0.924	0.862
0.7	1.5	10	75	0.059	0.021	0.031	0.808	0.924	0.862
0.7	1.7	10	75	0.059	0.021	0.031	0.808	0.924	0.862
0.7	2	10	75	0.059	0.021	0.031	0.808	0.924	0.862
0.8	1.5	10	75	0.059	0.021	0.031	0.808	0.924	0.862
0.8	1.7	10	75	0.059	0.021	0.031	0.808	0.924	0.862
0.8	2	10	75	0.059	0.021	0.031	0.808	0.924	0.862
1	1.5	10	75	0.059	0.021	0.031	0.808	0.924	0.862
1	1.7	10	75	0.059	0.021	0.031	0.808	0.924	0.862
1	2	10	75	0.059	0.021	0.031	0.808	0.924	0.862
0.6	1.5	10	75.385	0.063	0.021	0.032	0.809	0.929	0.865
0.6	1.7	10	75.385	0.063	0.021	0.032	0.809	0.929	0.865
0.6	2	10	75.385	0.063	0.021	0.032	0.809	0.929	0.865
0.5	1.5	4	80.769	0	0	0	0.811	1	0.896
0.5	1.7	4	80.769	0	0	0	0.811	1	0.896
0.5	2	4	80.769	0	0	0	0.811	1	0.896
0.6	1.5	4	80.769	0	0	0	0.811	1	0.896
0.6	1.7	4	80.769	0	0	0	0.811	1	0.896

0.6	2	4	80.769	0	0	0	0.811	1	0.896
0.5	1.5	0	81.154	0	0	0	0.812	1	0.896
0.5	1.7	0	81.154	0	0	0	0.812	1	0.896
0.5	2	0	81.154	0	0	0	0.812	1	0.896
0.6	1.5	0	81.154	0	0	0	0.812	1	0.896
0.6	1.7	0	81.154	0	0	0	0.812	1	0.896
0.6	2	0	81.154	0	0	0	0.812	1	0.896
0.7	1.5	0	81.154	0	0	0	0.812	1	0.896
0.7	1.5	4	81.154	0	0	0	0.812	1	0.896
0.7	1.7	0	81.154	0	0	0	0.812	1	0.896
0.7	1.7	4	81.154	0	0	0	0.812	1	0.896
0.7	2	0	81.154	0	0	0	0.812	1	0.896
0.7	2	4	81.154	0	0	0	0.812	1	0.896
0.8	1.5	0	81.154	0	0	0	0.812	1	0.896
0.8	1.5	4	81.154	0	0	0	0.812	1	0.896
0.8	1.7	0	81.154	0	0	0	0.812	1	0.896
0.8	1.7	4	81.154	0	0	0	0.812	1	0.896
0.8	2	0	81.154	0	0	0	0.812	1	0.896
0.8	2	4	81.154	0	0	0	0.812	1	0.896
1	1.5	0	81.154	0	0	0	0.812	1	0.896
1	1.5	4	81.154	0	0	0	0.812	1	0.896
1	1.7	0	81.154	0	0	0	0.812	1	0.896
1	1.7	4	81.154	0	0	0	0.812	1	0.896
1	2	0	81.154	0	0	0	0.812	1	0.896
1	2	4	81.154	0	0	0	0.812	1	0.896

Table 5.4.1: Measure values for no stemming

The accuracy as expected decreases with increase in the vocabulary cut off values, but as can be seen the best accuracy has zero measures for class one. Thus the best precision, recall and f-score should determine the best parameters. The reason for not having picked parameters for vocabulary cutoff and resizing using the validation data was the confusion between the performances in each case. For some classifiers a different parameter works, and in some cases, what is best for class one is not the best for class two. From this point onwards we pick both validation and testing was done with only vocabulary cutoffs for 50 and 100 and vocabulary resizing 1 and 2. Though the vocabulary resizing ratios did not show any difference in performance for any of its values.

Stemming	Vocab resize Ratios	Vocab Cut off at	Accuracy (%)	Precision of class one	Recall of class one	F-score of class one	Precision of class two	Recall of class two	F-score of class two
None	All	0/4	81.154	0	0	0	0.812	1	0.896
	All	100	58.462	0.174	0.1	0.127	0.804	0.886	0.843
Lovins Stemmer	All	50	62.308	0.261	0.267	0.264	0.82	0.815	0.817
	All	0	77.692	0.2	0.061	0.094	0.812	0.943	0.873

Table 5.4.2: Shows the best results (accuracy and best measures for class one) for no stemming and Lovins stemmer

The performance of the Stemmer is definitely better than no stemming in that, it increases the accuracy and also the measures of class one. The increase in accuracy and precision was lesser for the validation data set.

## 5.5 IB-1 (Instance Based – learning algorithms)

There are three different versions of this Nearest-neighbor classifier that were attempted. His algorithm uses normalized Euclidean distance to find the training instance closest to the given test instance, and predicts the same class as this training instance. If multiple instances have the same (smallest) distance to the test instance, the first one found is used. Instance based learning algorithms are similar to edited nearest neighbor algorithms but their primary purpose is to maximize classification accuracy (on novel instances if necessary) and are thus incremental. Thus they allow for better efficiency on real-world data with noise.

In addition to a distance function used to measure the similarity between a test instance and a testing instance the Instance Based learning algorithms have a concept description updater which determines whether new instances should be added to the instance database and which instances from the database should be used in classification.

No Stemming involved									
Vocab resize between		Vocab cut off at	Accuracy (%)	Precision of class one	Recall of class one	F-score of class one	Precision of class two	Recall of class two	F-score of class two
1	2	50	75	0.3	0.245	0.27	0.832	0.867	0.849
1	2	100	76.154	0.333	0.265	0.295	0.837	0.877	0.856
Lovins Stemmer									
1	2	50	75.385	0.317	0.265	0.289	0.836	0.867	0.851
1	2	100	76.923	0.351	0.265	0.302	0.839	0.886	0.862

The accuracy levels with or without stemming are good enough and the precision and recall values for class one are greater than 0.3 though not as close to class two's values. These precision and recall values lead to the idea that increase the nearest neighbors number might provide a

## 5.6 IB-5

This is the same as the above Instance based learning algorithm with number of nearest neighbors set as five. As hypothesized increasing the number of neighbors to five gave an improved accuracy with and without a stemmer. But the precision and recall for class one dropped in the no stemming case. In the case of Lovins Stemmer, for class one precision increases whereas recall and F-score decreases. For class two precision decreases with or without stemming and recall and F-score increases.

No Stemming involved									
Vocab resize between		Vocab cut off at	Accuracy (%)	Precision of class one	Recall of class one	F-score of class one	Precision of class two	Recall of class two	F-score of class two
1	2	50	78.4615	0.231	0.061	0.097	0.814	0.953	0.878
1	2	100	78.4615	0.348	0.163	0.222	0.827	0.929	0.875
Lovins Stemmer									
1	2	50	80.3846	0.429	0.122	0.19	0.825	0.962	0.888
1	2	100	81.5385	0.526	0.204	0.294	0.838	0.957	0.894

## 5.7 IB-10

In this case, the number of nearest neighbors is set to ten. Increasing the number of nearest neighbors still does increase the accuracy but the precision and recall still follows the same trends of increasing for one class and decreasing for the other.

No Stemming involved									
Vocab resize between		Vocab cut off at	Accuracy (%)	Precision of class one	Recall of class one	F-score of class one	Precision of class two	Recall of class two	F-score of class two
1	2	50	79.615	0.333	0.082	0.131	0.819	0.962	0.885
1	2	100	81.539	0.538	0.143	0.226	0.83	0.972	0.895
Lovins Stemmer									
1	2	50	80.385	0.333	0.041	0.073	0.815	0.981	0.89
1	2	100	82.692	0.7	0.143	0.237	0.832	0.986	0.902

## 5.8 Classification via Regression

Binary classification can also be dealt with as regression. There are a few advantages in doing this as it leads to more uniform convergence and computational requires lesser calls to the regressor. As was expected the precision of class one while using the stemmer is much higher than any other classifier has obtained and is also in the same range as that of class two.

No Stemming involved									
Vocab resize between		Vocab cut off at	Accuracy (%)	Precision of class one	Recall of class one	F-score of class one	Precision of class two	Recall of class two	F-score of class two
1	2	50	80.7692	0.4	0.041	0.074	0.816	0.986	0.893
1	2	100	80.7692	0.429	0.061	0.107	0.818	0.981	0.892
Lovins Stemmer									
1	2	50	82.6923	0.75	0.122	0.211	0.829	0.991	0.903
1	2	100	81.5385	0.6	0.061	0.111	0.82	0.991	0.897

## 5.9 Support Vector Machines

Uses the wrapper provided by WEKA for LIBSVM. The default arguments are used:

1. Type of SVM : C-SVC
2. Type of Kernel Function : radial basis function:  $\exp(-\gamma * |u-v|^2)$
3. Degree in kernel function : 3
4. Parameter C for C-SVC is 1

No Stemming involved									
Vocab resize between		Vocab cut off at	Accuracy (%)	Precision of class one	Recall of class one	F-score of class one	Precision of class two	Recall of class two	F-score of class two
1	2	50	81.15	0	0	0	0.812	1	0.896
1	2	100	81.15	0	0	0	0.812	1	0.896
Lovins Stemmer									
1	2	50	81.15	0	0	0	0.812	1	0.896
1	2	100	81.15	0	0	0	0.812	1	0.896

The performance of SVM is very disappointing considering that it is a maximum-margin classifier. The measures for class one is totally zero. It follows the hypothesis we had for Naïve bayes classifier saying that it blindly classifies everything as class two. The confusion matrix had classified all of the test instances as class two and of them only 41 turned out to be wrong and thus the accuracy continues to be good even though the classification is almost absent.

## 5.10 K-Star

K\* is an instance-based classifier, where the class of a test instance is based upon the class of those training instances similar to it, as determined by some similarity function. The underlying assumption of instance-based classifiers such as K\*, IB1, PEBLS, etc, is that similar instances will have similar classes. The corresponding components of an instance-based learner are the distance function which determines how similar two instances are, and the classification function which specifies how instance similarities yield a final classification for the new instance.

Nearest neighbor algorithms are the simplest of instance-based learners. They use some domain specific distance function to retrieve the single most similar instance from the training set. The classification of the retrieved instance is given as the classification for the new instance. The K-star algorithm [12] computes the distance between two instances based on information theory. The intuition is that the distance between instances be defined as the complexity of transforming one instance into another. The calculation of the complexity is done in two steps. First a finite set of transformations which map instances to instances is defined. A “program” to transform one instance (*a*) to another (*b*) is a finite sequence of transformations starting at *a* and terminating at *b*.

No Stemming involved									
Vocab resize between		Vocab cut off at	Accuracy (%)	Precision of class one	Recall of class one	F-score of class one	Precision of class two	Recall of class two	F-score of class two
1	2	50	82.3077	0.636	0.143	0.233	0.831	0.981	0.9
1	2	100	80.3846	0.375	0.061	0.105	0.817	0.976	0.89
Lovins Stemmer									
1	2	50	81.1538	0.5	0.041	0.075	0.816	0.991	0.895
1	2	100	81.1538	0.5	0.041	0.075	0.816	0.991	0.895

K-star algorithm was expected to perform better than the IB-k classifiers because of its distance calculation function, but though the accuracy definitely are similar and the precision values are similar in comparison to IB-1, the recall and F-score for class one with and without stemming are not good. The same was not true with the validation set. The performance there was more promising than it turned out with the test set.

## 5.11 LWL (Locally weighted learning)

This algorithm uses an instance-based algorithm to assign instance weights. There are five different options to set the weighting kernel shape to. All the five of them were attempted (Linear, Epanechnikov, Tricube, Inverse, Gaussian.) but all of them gave results similar to SVM. And seem to follow the same hypothesis of blindly classifying everything as class two.

No Stemming involved									
Vocab resize between		Vocab cut off at	Accuracy (%)	Precision of class one	Recall of class one	F-score of class one	Precision of class two	Recall of class two	F-score of class two
1	2	50	81.15	0	0	0	0.812	1	0.896
1	2	100	81.15	0	0	0	0.812	1	0.896
Lovins Stemmer									
1	2	50	81.15	0	0	0	0.812	1	0.896
1	2	100	81.15	0	0	0	0.812	1	0.896

## 5.12 Bayes Net

Bayes Networks are a powerful classification method even when there is incomplete data or some parameters are not present. The performance in the test set was better than the validation set. The precision and recall values for class one are if not the best so far, they are average and consistent irrespective of vocabulary cut off value and stemming.

No Stemming involved									
Vocab resize between		Vocab cut off at	Accuracy (%)	Precision of class one	Recall of class one	F-score of class one	Precision of class two	Recall of class two	F-score of class two
1	2	50	80	0.412	0.143	0.212	0.827	0.953	0.885
1	2	100	80	0.412	0.143	0.212	0.827	0.953	0.885
Lovins Stemmer									
1	2	50	80	0.412	0.143	0.212	0.827	0.953	0.885
1	2	100	80	0.412	0.143	0.212	0.827	0.953	0.885

## 5.13 Comparison

### 5.13.1 Results without Stemming

From Figure 5.13.1.1 we can see that the maximum accuracy is achieved by IB-10 and the minimum by Classification via clustering.

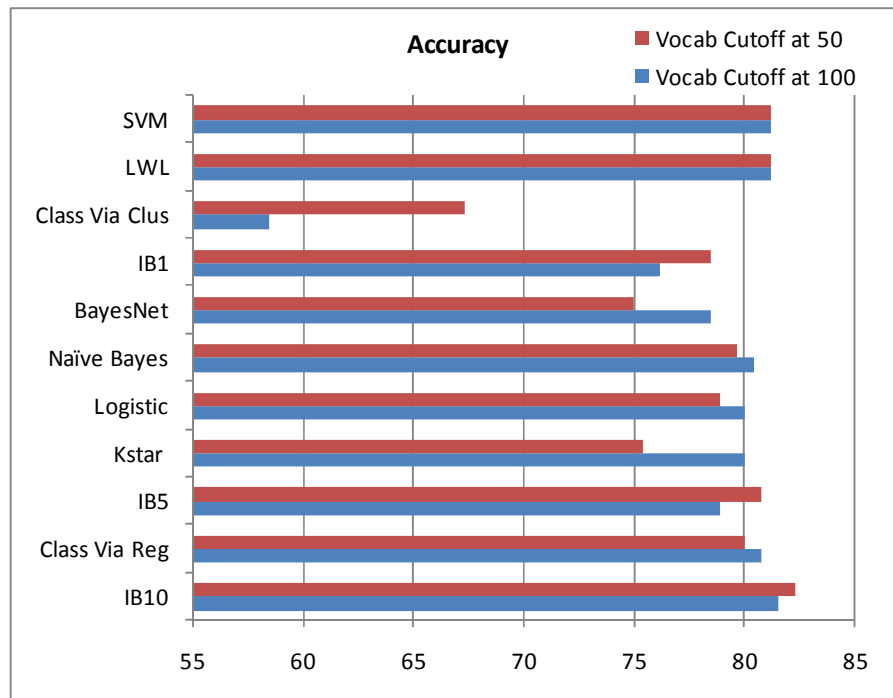


Figure 5.13.1.1: Accuracy in Percentage

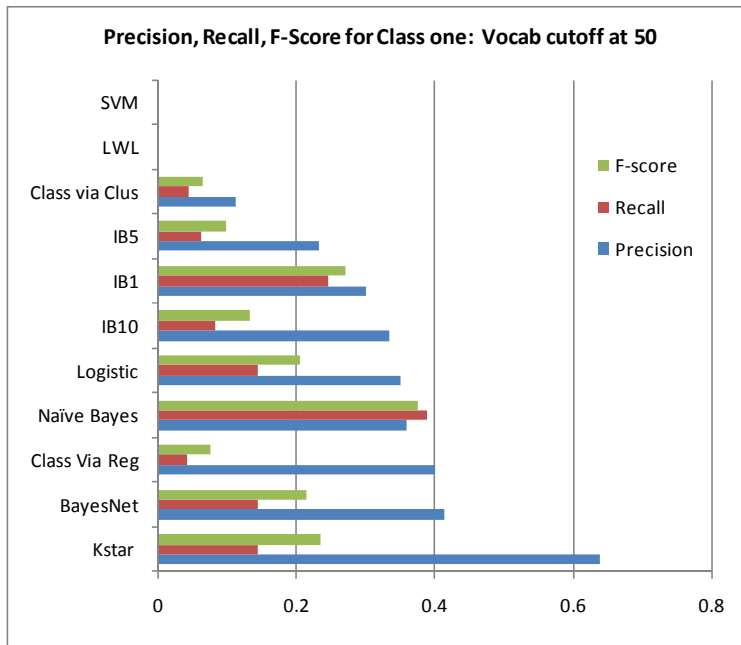


Figure 5.13.1.2: Measures for class one with vocab cut off at 50

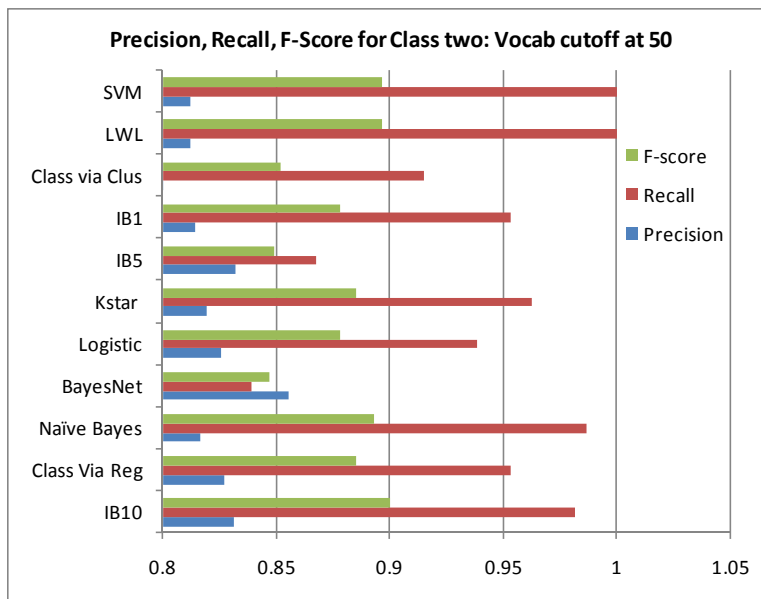


Figure 5.13.1.3: Measures for class two with vocab cut off at 50

The maximum precision is achieved by K-Star and maximum recall and F-score is by Naïve Bayes. The least recall, precision and f-score is by Classification via regression and clustering when the vocab cut off is at 50 and class one. For class two, maximum precision is by Naïve Bayes, and maximum f-score IB-10 and maximum recall is by SVM and LWL. Least precision is by SVM and LWL. Least recall and F-score is by Bayes Net for vocabulary cut off at 50.

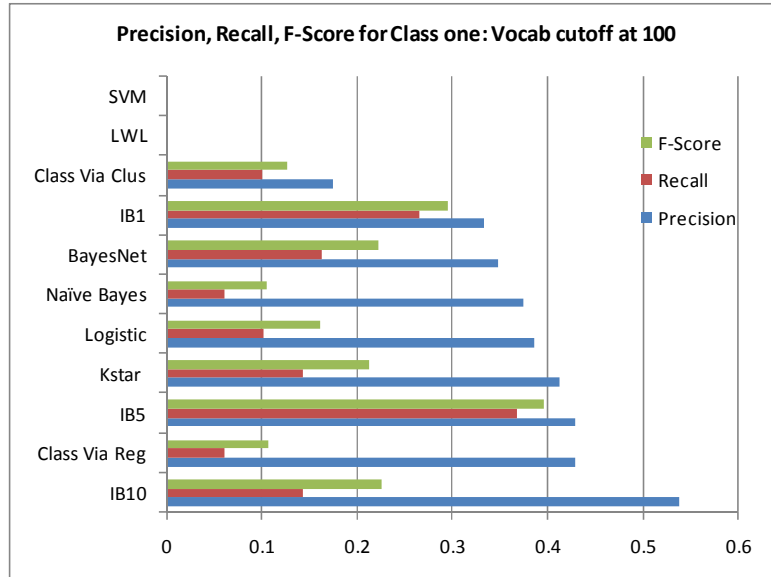


Figure 5.13.2.4: Measures for class one with vocab cut off at 100

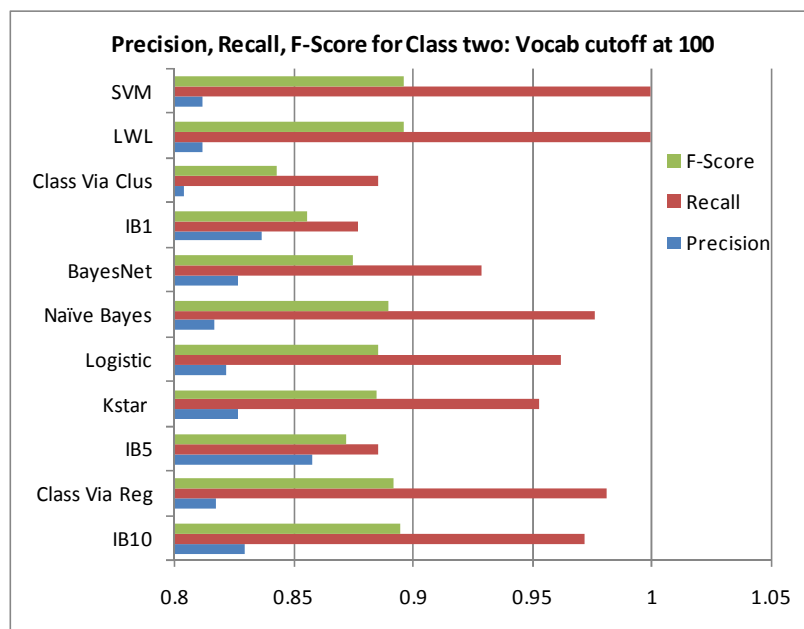


Figure 5.13.2.5: Measures for class two with vocab cut off at 100

. The maximum precision is achieved by IB-10 and maximum recall and F-score is by IB-5. The least recall, precision and f-score is by SVM and Locally Weighted learning when the vocab cut off is at 50 and class one. For class two, maximum precision is by IB-5, maximum recall and f-score is by SVM and LWL. Least precision, recall and F-score is by Classification via clustering for vocabulary cut off at 50.

### 5.13.2 Results with Stemming

From Figure 5.13.2.1 we can see that the maximum accuracy is achieved by IB-10 and the minimum by Classification via clustering.

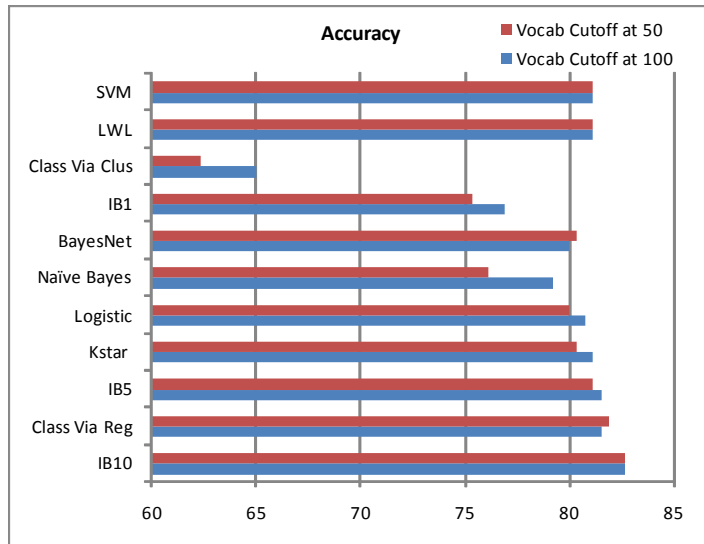


Figure 5.13.2.1: Accuracy in Percentage

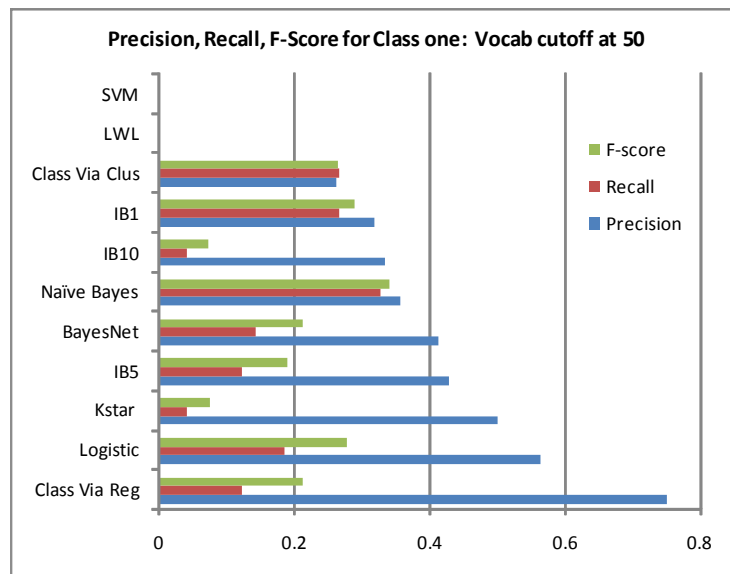


Figure 5.13.2.2: Measures for class one with vocab cut off at 50

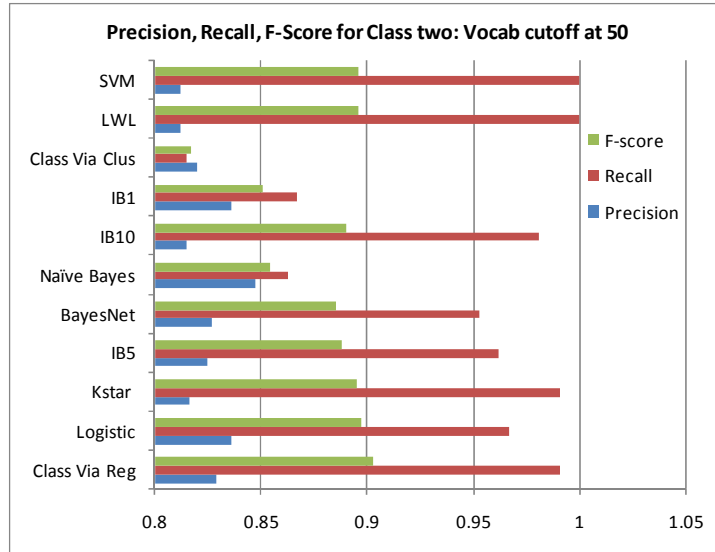


Figure 5.13.2.3: Measures for class two with vocab cut off at 50

The maximum precision is achieved by Classification via Clustering and maximum recall and F-score is by Naïve Bayes. The least recall, precision and f-score is by SVM and Locally Weighted learning when the vocab cut off is at 50 and class one. For class two, maximum precision and f-score is by Naïve Bayes, maximum recall is by SVM and LWL. Least precision is by SVM and LWL. Least recall and F-score is by Classification via clustering for vocabulary cut off at 50.

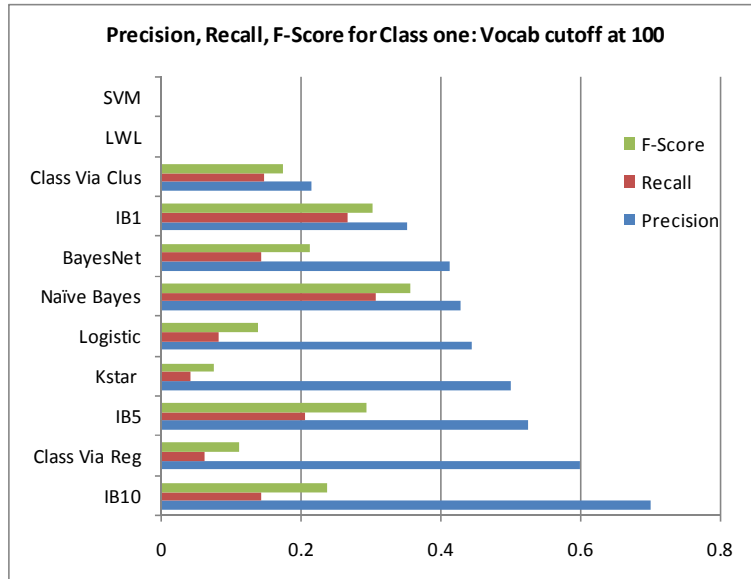


Figure 5.13.2.4: Measures for class one with vocab cut off at 100

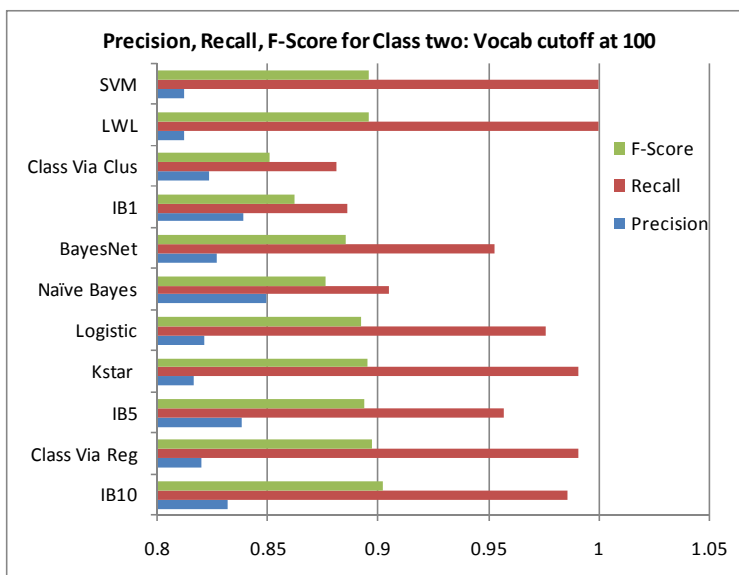


Figure 5.13.2.5: Measures for class two with vocab cut off at 100

The maximum precision is achieved by IB-10 and maximum recall and F-score is by Naïve Bayes. The least recall, precision and f-score is by SVM and Locally Weighted learning when the vocab cut off is at 50 and class one. For class two, maximum precision is by Naïve Bayes, maximum recall is by SVM and LWL and maximum F-score is by IB-10. Least precision is by SVM and LWL. Least recall and F-score is by Classification via clustering for vocabulary cut off at 50.

## 6 Maximum Entropy Classification

The Stanford Classifier, which is a Java Implementation of the conditional log-linear model classification (a.k.a. maximum entropy or multiclass logistic regression models). A number of features were tried for the problem at hand. For the training instances, only those words that were tagged as adjectives were used, instead of using the tag of the word as a feature. But, the previous tags and next word tags were also attempted to check if they increased performance.

FEATURES SET	
A_COUNT	NEXT_NEXT_NEXT_TAG
BUSINESS_ADDRESS	NEXT_NEXT_NEXT_WORD
BUSINESS_CATEGORIES	NEXT_NEXT_TAG
BUSINESS_NAME	NEXT_NEXT_WORD
BUSINESS_TAGS	NEXT_TAG
BUSINESS_URL	NEXT_WORD
CHAR_CAP_COUNT	NUM_CATEGORIES
CHAR_CAP_COUNT	NUM_ON_END
CHAR_NUMBER_COUNT	NUM_TAGS
COMMA	O_COUNT
CONSONANT_COUNT	PREV_LABEL
CONSONANT_VOWEL_RATIO	PREV_PREV_PREV_PREV_TAG
CONTINUOUS_CAP_COUNT	PREV_PREV_PREV_PREV_WORD
CONTINUOUS_NUMBER_COUNT	PREV_PREV_PREV_TAG
DASH	PREV_PREV_PREV_WORD
DASH_COUNT	PREV_PREV_PREV_WORD
E_COUNT	PREV_PREV_TAG

FIRST_CHAR_CAP	PREV_PREV_WORD
FIRST_WORD	PREV_TAG
HAS_CAP	PREV_WORD
HAS_NUMBER	REVIEW_OPINIONS
I_COUNT	U_COUNT
LENGTH	VOWEL_COUNT
LENGTH_TAGS	WORD
LOWERCASE	WORD_POSITION
NEXT_NEXT_NEXT_NEXT_TAG	WORD_POSITION_NORM
NEXT_NEXT_NEXT_NEXT_WORD	Y_COUNT

Table 6.1: All the features that were attempted. (54)

Table 6.1 lists all the features that were attempted on the validation set. After a rigorous hill climbing procedure the features set was narrowed down to Table 6.2. Classification was done according to variation type V3 and V1. First V3 results are shown and then V1 results. The best of the features were chosen by using the sum of the f-scores for each class as a final measure.

<b>Best feature Set</b>
<b>Score: 1.11686</b>
<b>Accuracy: 75.09%</b>
A_COUNT
BUSINESS_NAME
CHAR_CAP_COUNT
CHAR_NUMBER_COUNT
COMMA
CONTINUOUS_NUMBER_COUNT
DASH
E_COUNT
FIRST_CHAR_CAP
HAS_CAP
HAS_NUMBER
I_COUNT
LENGTH_TAGS
NEXT_NEXT_NEXT_NEXT_WORD
NEXT_NEXT_NEXT_TAG
NEXT_NEXT_NEXT_WORD
NEXT_NEXT_TAG
NEXT_TAG
NEXT_WORD
NUM_ON_END
PREV_LABEL
PREV_PREV_PREV_PREV_WORD
PREV_PREV_TAG
PREV_PREV_WORD
PREV_WORD
REVIEW_OPINIONS
SHORTLENGTH
VOWEL_COUNT
WORD
WORD_POSITION
WORD_POSITION_NORM
Y_COUNT

Table 6.2: The best set has 33 features

Without using any features other than what comes externally with the review, like Business categories name and tags information etc., it can be seen that the score is lesser than the score obtained using other features like the vowel count, length of the word, etc.

<b>Maximum Entropy Classifier</b>	<b>79.81%</b>	1.017977
	75.09%	<b>1.11686</b>

Table 6.3

Only Business Details as features
<b>Score: 1.017977</b>
<b>Accuracy: 79.81%</b>
BUSINESS_NAME
BUSINESS_TAGS
BUSINESS_URL
BUSINESS_CATEGORIES
BUSINESS_ADDRESS
LENGTH_TAGS
NUM_CATEGORIES
NUM_TAGS
PREV_LABEL
REVIEW_OPINIONS
WORD

Table 6.4: Only business details as features

Comparing these results to the other classifiers, we can see that the performance is close enough to be similar. Maybe exploring more features might increase its performance.

While running the same classifier for Variation V1, some of the previous classifiers were also run in a similar way to perform a comparison between them and MaxEnt classifier. It was hoped that the MaxEnt would perform better in this case than the other classifiers. The best score reached in the case of the other classifiers was 1.62 whereas the best with the MaxEnt classifier was 1.20. This was disappointing considering that the Maximum Entropy model was expected to perform better. But the possible reason for the performance might be the classification variation used.

Classifier	Stemming	Resizing between	Vocab cutoff	Accuracy	Final Score
K-star	0	1 2	50	45.4	<b>1.62</b>
IB1	0	1 2	100	44.2	<b>1.58</b>
Naïve Bayes	0	1 1.7	0	41.2	1.57
IB1	1	1 2	50	42.7	1.57
Naïve Bayes	0	1 1.5	0	40.8	1.5
K-star	1	1 2	50	46.5	1.5
IB1	1	1 2	100	43.8	1.48
IB1	0	1 2	50	41.9	1.47
Naïve Bayes	0	0.5 1.5	100	39.2	1.45
Naïve Bayes	0	0.5 1.7	100	39.2	1.45
Naïve Bayes	0	0.5 2	100	39.2	1.45
K-star	0	1 2	100	46.5	1.42
IB5	1	1 2	100	46.2	1.41

Naïve Bayes	0	0.5	1.7	50	37.3	1.41
Naïve Bayes	0	0.5	2	50	37.3	1.41
IB5	0	1	2	100	40.8	1.39
Logistic	0	1	2	50	45.4	1.36
K-star	1	1	2	100	46.2	1.36
IB10	1	1	2	100	49.2	1.34
IB5	0	1	2	50	43.5	1.31

Table 6.5: The best measures for other classifiers for Variation V1 (0 is for no stemming and 1 for Lovins stemmer)

Maximum Entropy Classifier	44.2308%	1.050082
	42.3%	1.200364

Table 6.6: Best of the Maximum entropy results for variation 1

## 7 Conclusion

The best accuracy achieved with machine learning classifiers was 82.307% with K-star algorithm using No stemming, Vocabulary count cutoff at 50. With stemming, it was 82.6923% with IB-10, vocabulary count cutoff at 100 and vocabulary resize between 1 and 2. When performance is measured as the sum of F-scores, the best score with no stemming was Naïve Bayes with 78.8462% accuracy and 1.268 as the score and with stemming it was Naïve Bayes again with 79.2308% accuracy and 1.233 as the score.

In the case of Maximum Entropy classifier case, classification by variation 3 gave maximum accuracy of 79.81% with just the business details as features (including the word and the previous word label that was predicted) and a maximum F-score of 1.11686 as shown in Section 6.

	Classifier	Accuracy (%)	F-score
No Stemming	K-Star	82.307	1.133
	IB-10	78.8462	1.268
Stemming	Naïve Bayes	82.6923	1.139
	Naïve Bayes	79.2308	1.233
Maximum Entropy Classifier		79.81%	1.017977
		75.09%	1.11686

Table 7.1: Final Results (MaxEnt results for variation 3)

## 8 Future Work

The performance can be tried to improve by exploring the following options:

1. Use N-gram models
2. Attempt more complex classifiers
3. Try Sentence Boundary options
4. Explore some other POS taggers

Some of the reviews contain sentences like, “ I was in a very good mood but the food was disappointing” contain conflicting adjectives, even though only one adjective “disappointing” is relevant to the review, the above method of classification gives the same treatment to both adjectives, which leads to a higher probability for error. Some method to overcome this could be explored by using N-gram models.

Some of the reviews contain sentences with conflicting opinions like, "The chicken was awesome and the atmosphere very calming. When I moved on to try the pancakes, they left a bad taste and the waiters are not very friendly." The first line is a positive review sentence while the second sentence carries a negative review. This leads to confusion when trying to settle upon the rating for this review. Changing the approach to consider each sentence rather than the whole review at a time might be better. Each sentence's tone can be identified and some kind of averaging function can be used to settle on the final rating of the review.

## 9 References

[1] M. A. Hearst, "Direction-based text interpretation as an information access refinement " in Text-based intelligent systems: current research and practice in information extraction and retrieval Lawrence Erlbaum Associates, Inc., 1992 pp. 257-274

[2] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment Classification Using Machine Learning Techniques," in Proceedings of EMNLP 2002

[3] P. D. Turney, "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Un-supervised Classification of Reviews," in Proceedings of the 40th ACL, 2002, pp. 417-424.

[4] P. Subasic and A. Huettner, "Affect analysis of text using fuzzy semantic typing," Fuzzy Systems, IEEE Transactions on, vol. 9, pp. 483-496, 2001.

[5] H. Liu, H. Lieberman, and T. Selker, "A model of textual affect sensing using real-world knowledge " in Proceedings of the 8th IUI, ACM Press, 2003 pp. 125-132

[6] M. Hu and B. Liu, "Mining and summarizing customer reviews " in Proceedings of the 2004 SIGKDD, 2004 pp. 168-177

[7] C. Fellbaum, WordNet: an Electronic Lexical Database: MIT Press, 1998.

[8] Web-harvest: <http://web-harvest.sourceforge.net/>

[9] Stanford POS Tagger: <http://nlp.stanford.edu/software/tagger.shtml>

[10] V. Hatzivassiloglou and K. R. McKeown, "Predicting the semantic orientation of adjectives" in Proc. of 8th EACL, 1997

[11] <http://weka.sourceforge.net/doc.dev/>

[12] John, G. Cleary and Leonard, E. Trigg (1995) "K\*: An Instance- based Learner Using an Entropic Distance Measure", *Proceedings of the 12th International Conference on Machine Learning*, pp. 108-114.