SENTIMENT ANALYSIS ON MOVIE REVIEWS: AN EVALUATION OF NAÏVE BAYES AND SVM FOR SENTIMENT CLASSIFICATION

Tajudeen Abdulazeez
IST738, Text Mining
School of Information, Syracuse University
toabdula@syr.edu

INTRODUCTION

Sentiment analysis is the type of text research that involved the use of statistics, natural language processing and machine learning to identify and extract subjective information from text, for instance, a reviewer's feelings, thoughts, judgments, or assessments about a topic, event, or a company and its activities as mentioned above. It can also be categorized under a predictive text mining for text categorization.

Sentiment analysis can be broadly categorized into two depending on the scale:

1. Coarse-grained sentiment analysis
2. Fine-grained sentiment analysis

Corse-grained sentiment analysis is done on a document and sentence levels. This entails two coherent tasks: Subjective classification and sentiment detection and classification.

Subjective classification: It first determined if a sentence or document is subjective or objective. An Objective sentence contains some fact about an object or topic.

Sentiment detection and classification: The goal here is to first determine if a sentence has sentiment of not, if it does, to determine the emotion if it is positive, negative or neutral.

Fine-grained sentiment analysis is analyzing sentences by part and each part is analyzed in connection to others. The sentences are broken down into phrases. This analysis is carried out at sub-sentence levels and it is meant to identify a target topic of a sentiment.

The massive amount of data in social media platforms is a key source for companies to analyze customer sentiment and opinions. Many existing sentiment analysis approaches solely rely on textual contents of a sentence (e.g. words) for sentiment identification. Consequently, current sentiment analysis systems are ineffective for analyzing contents in social media because people may use non-standard language (e.g., abbreviations, misspellings, emoticons or multiple languages) in online platforms. (Fan, Ilk, & Zhang, 2015).

This experiment will explore the accuracy of both Multinomial Naïve Bayes and SVM(Support Vector Machine) in classifying sentiment of a restaurant reviews. The best performing model will be used to predict the test sets of the Kaggle competition to evaluate the performance of the model on a new dataset.

ANALYSIS AND MODEL

The analysis is carried out using python programming language with the following packages: Panadas, Numpy, Matplotlib, Seaborn, Sklearn and Scipy.

About the Data

The Rotten Tomatoes movie review dataset is a corpus of movie reviews used for sentiment analysis, originally collected by Pang and Lee [1]. In their work on sentiment treebanks, Socher et al. [2] used Amazon's Mechanical Turk to create fine-grained labels for all parsed phrases in the corpus. This competition presents a chance to benchmark your sentiment-analysis ideas on the Rotten Tomatoes dataset. You are asked to label phrases on a scale of five values: negative, somewhat negative, neutral, somewhat positive, positive. Obstacles like sentence negation, sarcasm, terseness, language ambiguity, and many others make this task very challenging.

The details description of each label is shown below

Sentiment Label	Description
0	Negative
1	Somewhat negative
2	Neutral
3	Somewhat positive
4	Positive

The distribution of the sentiment label across all classes is shown below in the histogram

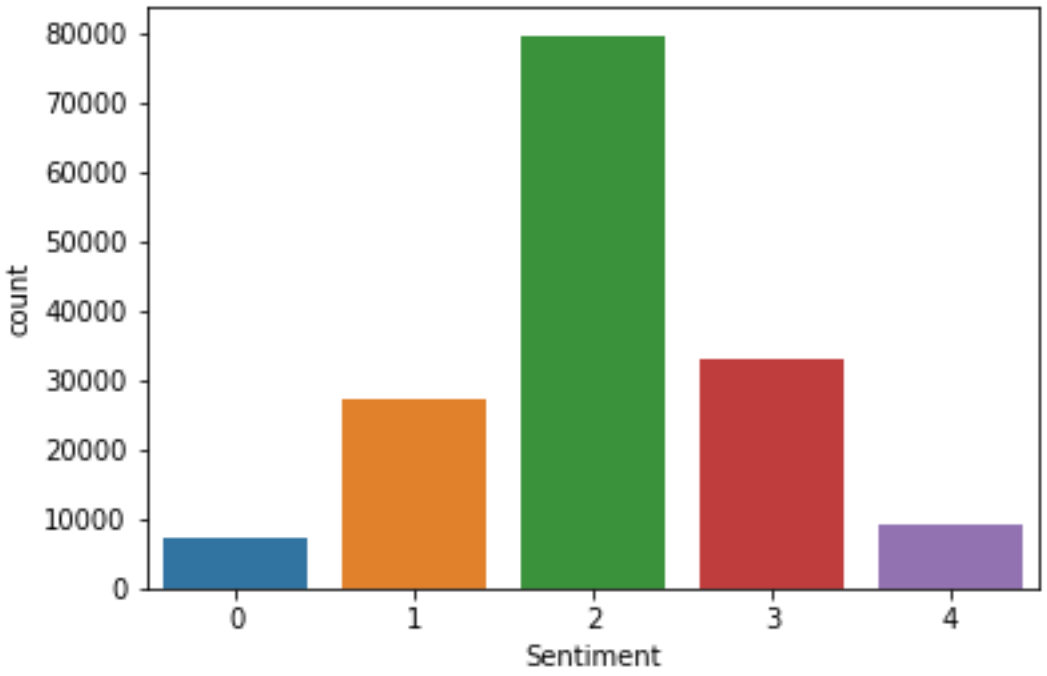

Figure 1: The distribution of the sentiment class, the neutral class has the majority count with about 50% of the datasets. The majority vote baseline accuracy is 51% using accuracy as the evaluation metrics.

Data Preparation

Sklearn CountVectorizer is used to transform the datasets into a vector that algorithms can work with. Different ngram number combination are tried to check the accuracy of different form of vectorization for both classifiers used for this analysis.

Let's get a little dive into what vectorization process means and the different types of vectorization.

Vectorization is the process of transforming text document in a way computer can understand. The only way a computer can understand text is to represent with number which is call text vectorization. This is an important step in data mining. Each word represented by a number is called a token.

Tokenization rules can vary depending on the business problems and how grouping of words can have significant impact in identifying trend in text document. The tokens can be represented with n-grams or bag of words. Bag of word tokens ignore the context of words while n-grams while n-grams capture the local content of words, the representation can be unigrams, token of individual words, bi-grams, token of two words, tri-grams, token of three words and so on.

There are four major different type of converting text to a vector for computer to understand.

1. Boolean Vectorizer
2. Count or Term Frequency Vectorizer
3. Normalized term Frequency
4. TF*IDF Vectorizer

Boolean is a vectorization method where each token is represented, either if it is present in a document or not. If a token is present, it is represented by 1 and 0 otherwise.

Term Frequency is a vectorization method where each token is represented by the number of occurrences in a document. If you a document and a token appeared in the document 5 times, the token or word will be represented by 5.

Normalize term frequency is a method of vectorization where word frequency is normalized by the document length. If you have a document with a length of 1000, and a word appeared in the document for 20 times, the normalized term frequency will be 20/1000.

Tf*idf is a concept borrowed from information retriever, and it's a blind weighting strategy for text classification. in this method, the term frequency is multiplying by the inverse document frequency. This is aimed to penalize the common word across documents.

Train/Test

The datasets are split into train test using skearn train_test_split into 70% to 30% proportion. 70% is used as the training sets while the 30% is used as the evaluation sets.

MODEL

Multinomial Naïve Bayes and SVM Classifier is used for this experiment and a brief description of both SVM and MNB (Multinomial Naïve Bayes) is follow:

SVM

An SVM is a supervised machine learning technique that uses support vectors in classification. The support vectors define the hyperplane which are the most important part of training an SVM.

Any line in dimension D can be represented as w T x + b = 0. w is a vector of coefficients, x is the vector of variables, b is the translation (can be thought of in 2D as the y intercept).

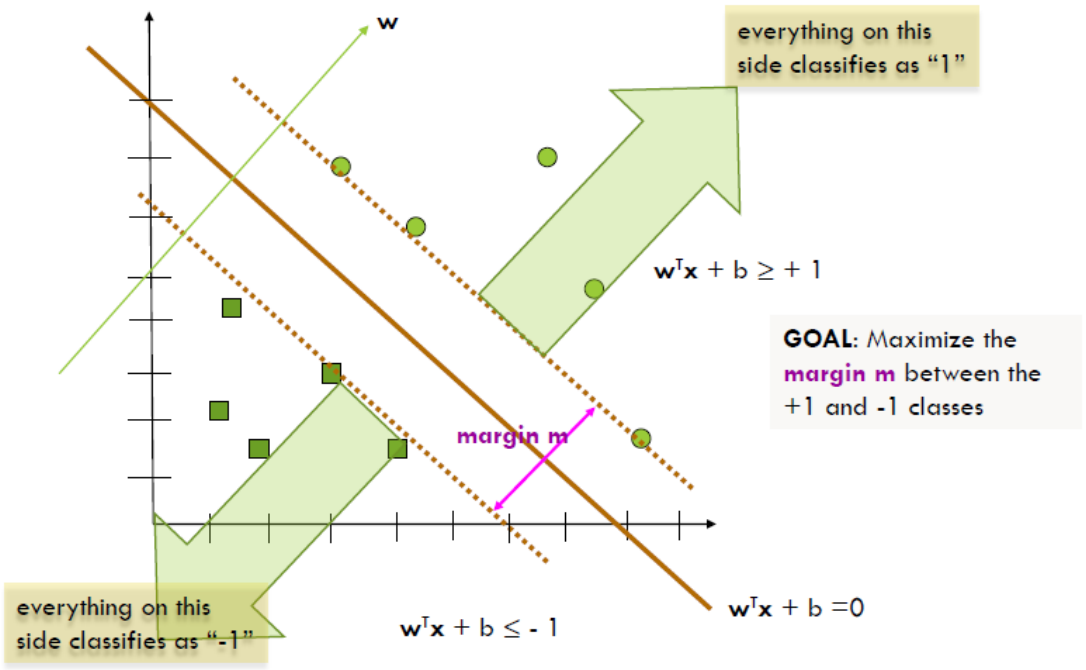

Figure 2: Setting up the SVM Problem. The goal is to maximize the margin m between the +1 and -1 classes.

It uses different "kernels " options such as linear, gamma, sigmoid, and Gaussian. SVM can perform feature transformation into higher (or infinite Hilbert Space) dimensional space so that input vectors (x) are separable by hyperplane (high dimensional plane).

SVM can be used for regression, classification and outlier detection and below are the summary of advantages of SVM:

1. Effective in high dimensional (D) space
2. Effective when dimension is greater than the number of samples
3. It uses subset of the training data, just the support vectors
4. Allows or the use of kernel functions

While some of the disadvantages are summarized below:

1. If the features is greater than the number of samples, there will be poor performance
2. No probability estimates- these must be done by hand using cross validation.

SVM did not required any special form of vectorization process. It will work fine with any form of text transformation compared to MNB that required a count vectorizer as input.

MNB (Multinomial Naïve Bayes)

MNB used the Bayes theorem assumption in classification problem. It assumed that all attributes are independent which make it suitable for text classification.

Naïve Bayes:

$$P(Y \mid X) = \frac{P(X,Y)}{P(X)}$$

$$P(X \mid Y) = \frac{P(X,Y)}{P(Y)}$$

$$P(Y \mid X) = \frac{P(X \mid Y)P(Y)}{P(X)}$$

Naïve Bayes is robust to isolated noise, handle missing values by ignoring instances during probability estimate calculations.

MNB required a count vectorizer as input for optimal performance.

Using Ngram Vectorizer

Using different Ngram combination and 10 fold cross validation for SVM and MNB for the sentiment classification. The accuracy result is shown in the table below:

	Model	Vectorization	10 fold Avg Score
0	SVM	ngram12	0.566385
1	SVM	ngram13	0.565923
2	SVM	ngram23	0.513418
3	SVM	ngram22	0.514148
4	MNB	ngram12	0.556632
5	MNB	ngram13	0.555491
6	MNB	ngram23	0.511508
7	MNB	ngram22	0.511630

Table 1: Accuracy results of 10 fold cross validation using different ngram combination.

SVM with ngram(1,2) perform better with an accuracy of 56.6% while the MNB perform better with ngram(1,2) with an accuracy of 55.6%.

MNB

The classification report of the MNB using the test sets is shown below:

	precision	recall	f1-score	support
0	0.32	0.40	0.36	1785
1	0.43	0.48	0.46	7335
2	0.75	0.70	0.73	26010
3	0.49	0.51	0.50	9469
4	0.35	0.43	0.39	2219
accuracy			0.60	46818
macro avg	0.47	0.50	0.49	46818
weighted avg	0.62	0.60	0.61	46818

Figure 3: The classification report of MNB using bigram vectorizer on the test set. The overall accuracy of the model is 60%. It performs better on predicting the positive class compared to the negative class.

Top 10 positive word Indicative	**Top ten negative word indicative**
meaningless vapid: 3.699299697094615 *devoid substance: 3.5367807675968415* *somewhat: 3.080622210377637* *definitely meaningless: 3.0061525165346694* *little puddle: 3.0061525165346694* *puddle: 3.0061525165346694* *puddle movie: 3.0061525165346694* *silly little: 3.0061525165346694* *vapid devoid: 3.0061525165346694* *hours precious: 2.9007920008768444*	*ballistic: 0.01042024298067723* *blonde: 0.01042024298067723* *chills: 0.01042024298067723* *considerable: 0.01042024298067723* *cross: 0.01042024298067723* *excited: 0.01042024298067723* *explosion: 0.01042024298067723* *farts urine: 0.01042024298067723* *generate: 0.01042024298067723* *gun: 0.01042024298067723*

Table 2: Top 10 indicative words for positive and negative sentiment using MNB and bigram as input.

SVM

Linear SVM with C=1. Below is the classification report on the test set using bigram as input.

	precision	recall	f1-score	support
0	0.36	0.48	0.41	1663
1	0.44	0.53	0.48	6703
2	0.82	0.71	0.76	27775
3	0.48	0.55	0.51	8458
4	0.40	0.49	0.44	2219
accuracy			0.64	46818
macro avg	0.50	0.55	0.52	46818
weighted avg	0.67	0.64	0.65	46818

Figure 4: The classification report of linear SVM on the test set using bigram as input. The overall accuracy is 64%.

Very Negative Word Indicative	Very Positive word indicative
(1.7328338247383934, 'stinks') *(1.7969055240622749, 'entirely witless')* *(1.8028028410319092, 'unwatchable')* *(1.865864210387101, 'disgusting')* *(1.8687740438101081, 'disappointingly')* *(1.8888413781376032, 'unappealing')* *(1.8946538601764895, 'turd')* *(1.8958294746216868, 'uninspiring')* *(1.8994515943283015, 'distasteful')* *(1.9658503400351988, 'pompous')*	*(1.6986938673796288, 'gem')* *(1.7015943926366317, 'flawless')* *(1.71335893661125, 'standout')* *(1.7952983829655527, 'masterpeice')* *(1.8203847591726836, 'amazing')* *(1.8342981386195634, 'cut rest')* *(1.893410116375843, 'miraculous')* *(1.9405575043367327, 'wo disappointed')* *(2.0156734905156233, 'masterpiece')* *(2.1419215835302703, 'perfection')*

Table 3: Top 10 positive and negative indicative words of linear SVC using bigram as input.

Unigram and Bigram

Be is the accuracy result using 10-fold cross validation of SVC and MNB using unigram and bigram vectors as input.

	Model	Vectorization	10 fold Avg Score
0	SVM	ngram12	0.566397
1	SVM	unigram	0.565968
2	MNB	ngram12	0.556632
3	MNB	unigram	0.561303

Table 4: Accuracy score of SVM and MNB using unigram and bigram as the input vectors.

MNB perform better using unigram vector compared to using bigrams.

The best performing model is still SVM using accuracy as the evaluation metrics based on the same types of input vectors.

Model Optimization

The best performing model (SVM) hyperparameters is tune to check for possible increase in model performance.

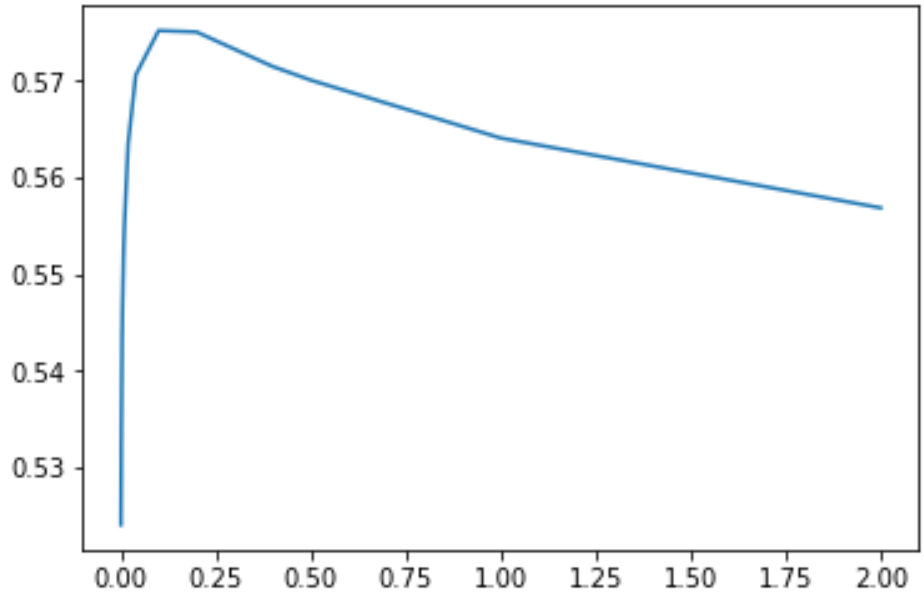

Figure 4: The x-axis is the C value; the y-axis is the accuracy score.

Different valued of C is tried, the value around 0.1 and 0.2 produce higher training accuracy but when tried on Kaggle test set, the accuracy of the model drop, an indication of overfitting.

The Value of C with the best accuracy score on the test set, using Kaggle test set is 1, with an accuracy 56%.

Feature Engineering

A new feature is created called 'has-neg'. This is used to identify some negation in the reviews with Boolean values. The accuracy of the model increases from 56% to 61% on Kaggle test.

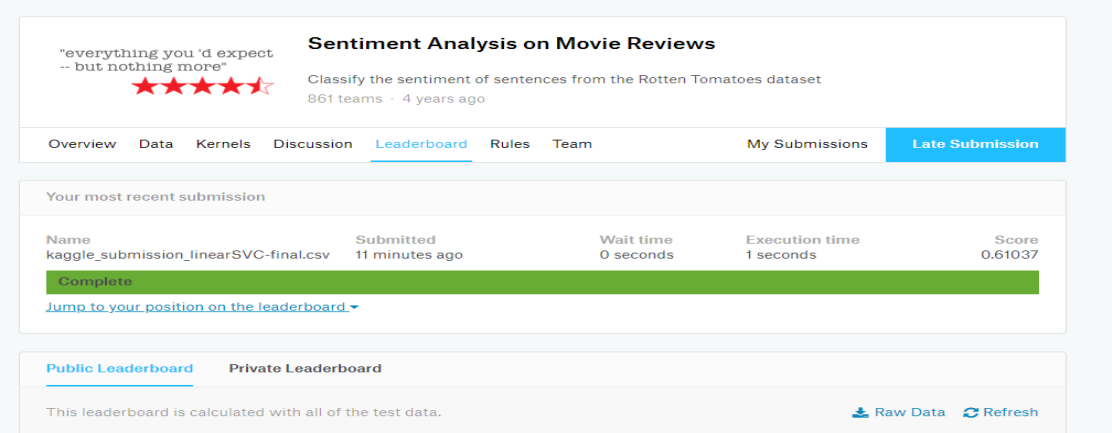

Figure 5: Accuracy score on Kaggle using the optimized SVM with feature engineering. 61%

Classification report of SVM

The classification report of the best performing SVM on the test set using C=1, bigram as input vectorizer and feature engineering as described above.

	precision	recall	f1-score	support
0	0.36	0.48	0.41	1663
1	0.44	0.53	0.48	6703
2	0.82	0.71	0.76	27775
3	0.48	0.55	0.51	8458
4	0.40	0.49	0.44	2219
accuracy			0.64	46818
macro avg	0.50	0.55	0.52	46818
weighted avg	0.67	0.64	0.65	46818

Figure 6: Classification report of the SVM on the test set with an accuracy of 64%.

Conclusion

We can make the following conclusion from the experimental results and analysis:

- Multinomial Naïve Bayes best performance in predicting the sentiment of the movie reviews using Count vectors as an input with an accuracy of 56%. The model exceeds the baseline accuracy of 50%.
- SVM best performance perform in predicting sentiment of movie reviews using bigram vectors as input with an accuracy of 61%. This exceeds the baseline accuracy of 50%.
- SVM perform better compared to MNB if the same type of input vectors is used in predicting movie reviews sentiment.
- SVM do well in classifying movie reviews sentiment using any types of vectorization as input.
- MNB perform better using count vectorization as input in predicting movie reviews sentiment.
- The accuracy score on Kaggle is 61% using SVM with C = 1 .