COMPARISION OF TWO SENTIMENT ANALYSIS ALGORYTHMS

Assoc. Prof. Atanassov A., Senior Lecturer Tomova F.
Department of Computer Science – University of Chemical Technology and Metallurgy, Bulgaria
naso@uctm.edu

Abstract: *This paper presents the comparison of the capabilities of two algorithms for Sentiment Analysis developed in Python. Both Python programs are used on the same Yelp dataset with customer reviews of the quality of the services in USA restaurants. The programs are based on open-source software frameworks and libraries as Python, NTLK, Scikit-Learn, Panda, etc. which are oriented to Machine and Learning and Natural Language Processing. The evaluation of the programs is based on precision of the predicted results and the compactness of the programming code. For model training and prediction, the Multinomial Naïve Bayes and Support Vectors Machines classifiers are applied in both algorithms.*

Keywords: MACHINE LEARNING; SENTIMENT ANALYSIS; NAÏVE BAES; SUPPORT VECTOR MACHINES; NLTK, NLP

1. Introduction

Sentiment Analysis /SA/, also known as opinion mining, is defined as the task of finding the opinions of authors about specific entities [5]. Sentimental analysis is used in various places: to analyze the reviews of a product whether they are positive or negative, to check if a political party campaign was successful or not, to analyze the reviews of a movie and to analyze the content of tweets or information on other social media [6]. Social media monitoring applications and companies depend on sentiment analysis and machine learning to assist them in gaining insights about mentions, brands, and products [7].

Sentiment Analysis refers to the use of Natural Language Processing /NLP/, text analysis and computational linguistics to identify and an extract subjective information from source materials [7]. SA is a process of exploring product reviews on the internet to determine the complete opinion. SA can be considered as a classification task as it sorts the location of a text into either positive or negative.

Sentiment Analysis can be considered a classification task as it classifies a text as positive or negative. Machine Learning /ML/ is one of the widely used approaches to analyzing opinion, and the most widely used for this purpose classification algorithms are based on Multinomial Naïve Bayes /MNB/ and Supported Vector Machines /SVM/ [1].

The work is a continuation of previous research [9,14] and presents a comparison of the precision of prediction of two mentioned above algorithms (MNB and SVM) developed in the Python. The algorithms are applied on Yelp dataset, available on kaggle.com.

2. Sentiment Analysis Algorithm

The SA algorithm is related to pre-processing of the original set of texts containing user opinions regarding a product or service, application of NLP [2] techniques to convert texts into numerical vectors to be processed by ML algorithms. Below are the main steps of this algorithm:

1. Retrieving the necessary recordings of opinions and associated attributes and characteristics as well as the label with the user's assessment and recording them in a database or table for the purpose of pre-processing and including meaningful or exclusion of certain unnecessary attributes in the relevant recordings.
2. Pre-processing of the text (word processing, tokenisation (words), punctuation, unions, identifying and non-limiting members, etc., reduction of words with the same root to the root, etc.)
3. Converting words into a numeric vector based on a dictionary of spoken words and the frequency of their occurrence in the text (Bag of Words /BoW/).
4. Application of Term Frequency-Inverse Document Frequency.
5. Forming a Sparse Matrix with rows equal to the number of words in the dictionary and with columns equal to the number of reviews.
6. Applying the classification algorithms based on the Multinomial Naïve Bayes and Support Vector Machines using 70% subset of data for training a model and 30% subset of data for testing the trained model.
7. Testing the model and assessing its precision.
8. Predicting the basis of the trained model.

2.1. Preprocessing text data

Some techniques and concepts of Natural Language Processing have been used to solve the above mentioned steps. (NLP) These are related to the following terms:

- ***Features:*** A feature represents an attribute or a property of an observation. It is also called a variable. A feature represents an independent variable. In a tabular dataset, a row represents an observation and column represents a feature. For example, consider a tabular dataset containing user profiles, which includes fields such as age, gender, profession, city, and so on. Each field in this dataset is a feature in the context of machine learning. Each row containing a user profile is an observation.

- ***Feature Extractors:*** Term frequency-inverse document frequency /TF-IDF/ is a feature vectorization method widely used in text mining to reflect the importance of words to a document in the amount.

Bag-of-Words /BoW/ is a representation of text that describes the occurrence of words within a document. In this method, each word count can be considering as a feature. Because ML algorithms cannot work with raw text data directly, the text must be converted into numbers. Exactly, vectors of numbers [3].

Feature Transformers

Tokenization is a transformer that converts an input string (text) to lowercase and splits it into words using whitespaces as a separator. A simple Tokenizer provides this functionality and splits sentences into sequences of words.

Stop Words Remover - stop words are words which should be excluded from the input because the words appear frequently and don’t carry as much meaning. Stop Words Remover takes as input a sequence of strings (output of a Tokenizer) and drops all the stop words from the input sequences [13].

2.2. Naïve Bayes Classification Algorithm

The Bayesian Classification represents a supervised learning method as well as statistical methods for classification. It can solve diagnostic and predictive problems. Naïve Bayes is a simple multiclass classification algorithm based on the application of Bayes’ theorem “The Bayes theorem is based on the concept of learning from experience that is, using a sequence of steps to come to a prediction. It is the calculation of probability based on prior knowledge of occurrences that might have led to the event” [6]. Naïve Bayes is a probabilistic model that makes predictions by computing the probability of a data point that goes to a given class [12]. Initially, the conditional probability distribution of each feature given class is computed, and then Bayes’ theorem is applied to predict the class label of an instance. Naive Bayes is used in a lot of practical real-life applications such as it is used in the sentimental analysis of text to classify the emotion of a particular piece of text, whether it is a positive sentiment or a negative one. This algorithm is fast to train and test; hence it is used in real-time prediction scenarios to make fast predictions on events based data that is generated in real time. It is used in many recommendation systems to give useful suggestions of content to the users.

2.3. Support Vector Machine (SVM) Algorithm

Support Vector Machine is another popular algorithm of the supervised machine learning algorithms that are used in many real life applications like text categorization, image classification, sentiment analysis and handwritten digit recognition. SVM is a powerful and popular technique for regression and classification. Unlike Naïve Bayes, it is not a probabilistic model but predicts classes based on whether the model evaluation is positive or negative [12]. SVM is used to classify the texts as positives or negatives. It works well for text classification due to its advantages such as its potential to handle large features [5].

3. *Used Machine Learning Libraries*

Programming language Python requires the **NLTK** platform (Figure 1) to build NLP code. The **NLTK** (Natural Language Toolkit) is a leading platform for building Python programs for working with data / texts in natural, human language. **NLTK** [8, 10] provides easy-to-use interfaces with over 50 corpus and lexical resources, such as WordNet, along with a set of word processing libraries for classification, tokenization, retrieval, tagging, analysis and semantic reasoning, NLP wrappers and other libraries.

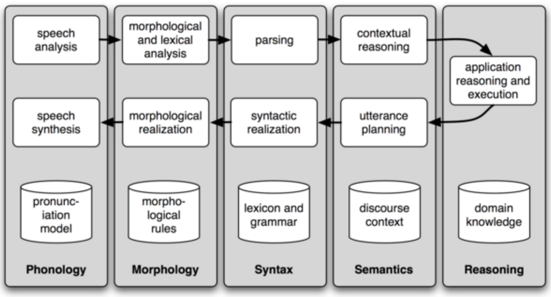


Fig. 1. *NLTK structure [4]*

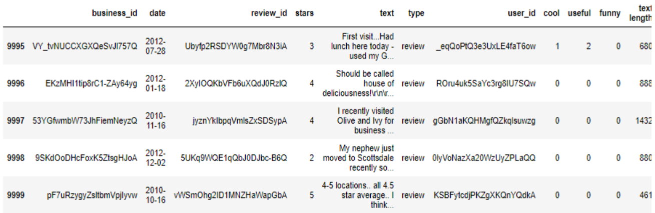
Scikit-learn in one of the most popular libraries for open-source Machine Learning for Python. It offers Machine Learning algorithms, including classification, regression, size reduction, and grouping. The library provides modules for data mining, data processing, and modeling. Scikit-learn is popular in academic research because it has a well-documented, easy-to-use and multifunctional user interface. Developers can use it to experiment with different algorithms by changing only a few lines of code. Scikit-learn has shells for some popular machine learning algorithms such as LIBSVM and LIBLINEAR. Other Python libraries, including NLTK, include scikit-learn wrappers. The library is packed with the most popular data sets, allowing developers to focus on algorithms rather than pre-processing and preparing data.

4. *Yelp Dataset*

Datasets are data pre-loaded with labels and prepared for use by supervised machine learning algorithms to analyze opinion. Numbers of data with labelled sentimental sentences (sentences containing sentiment, emotion, opinion) are available at www.kaggle.com.

Yelp dataset used in Python analysis programs include 10000 free-text reviews which are evaluating in five-stars the service in USA restaurants. Each entry (record) in the dataset contains the following columns (Figure 2):

business_id (business identifier)
date (day of review / review)
review_id (ID)
stars (1-5 rating for the company)
text (text to review)
type (type of text)
user_id (user ID)
cool/useful/funny-comments made by other users.



	business_id	date	review_id	stars	text	type	user_id	cool	useful	funny	text length
9995	VY_tvNUCCXGXQeSvJl757Q	2012-07-28	Ubyfp2RSDYW0g7Mbr8N3iA	3	First visit...Had lunch here today - used my G...	review	_eqQoPtQ3e3UxLE4faT6ow	1	2	0	680
9996	EKzMHI1tip8rC1-ZAy64yg	2012-01-18	2XyIOQKbVFb6uXQdJ0RzIQ	4	Should be called house of deliciousness!\r\n\r...	review	ROru4uk5SaYc3rg8IU7SQw	0	0	0	888
9997	53YGfwmbW73JhFiemNeyzQ	2010-11-16	jyznYkIbpqVmlsZxSDSypA	4	I recently visited Olive and Ivy for business ...	review	gGbN1aKQHMgfQZkqlsuwzg	0	0	0	1432
9998	9SKdOoDHcFoxK5ZtsgHJoA	2012-12-02	5UKq9WQE1qQbJ0DJbc-B6Q	2	My nephew just moved to Scottsdale recently so...	review	0lyVoNazXa20WzUyZPLaQQ	0	0	0	880
9999	pF7uRzygyZsltbmVpjIyvw	2010-10-16	vWSmOhg2ID1MNZHaWapGbA	5	4-5 locations.. all 4.5 star average.. I think...	review	KSBFytcdjPKZgXKQnYQdkA	0	0	0	461

Fig. 2. *Examples with the structure of the records of Yelp dataset*

Each record of Yelp dataset is labeled with customer estimation ranked between one end five stars where Star1 or Star2 means negative reviews, Star4 and Star5 are positive, and Star3 is neutral.

Next are examples with positive and negative reviews:

Positive review:

"Drop what you're doing and drive here. After I ate here I had to go back the next day for more. The food is that good." – Stars 5.

Negative review:

"I have always been a fan of Burlington's deals; however I will not be shopping at this one again. I went to customer service I think you should have some."– Stars 1.

5. *Developed SA Algorithms in Python*

5.1. Algorithms with Star1 and Star 5 subset of Yelp

With first two algorithms, based on MNB and Linear SVM, the data is initially processed by retrieving only the Star1 and Star5 scores that give meaning to positive (5) or negative (1) estimations. Thus, the dataset is reduced from 10000 to 4086 records.

```
yelp_class=yelp[(yelp['stars']==1)|(yelp['stars']
==5)]
yelp_class.shape
Output: (4086, 10)
```

On next figure (Fig. 3) the distribution of the records as the amount and length of their text is provided. As can be seen for most reviews the typical length is up to 1000 words and negative ones are shorter than positive ones.

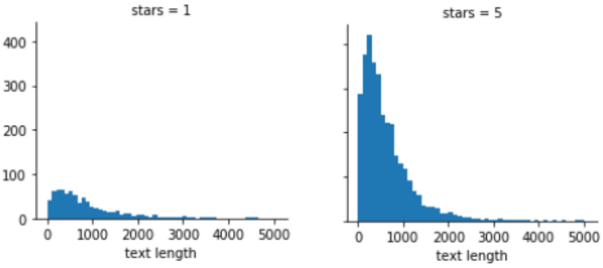


Fig. 3. *Text length distribution of positive and negative reviews*

On the reduced data (4086 entries), the text transformation operations described in step 2 using the NLTK library are applied. In the end, a vocabulary is formed - a vector of the unique words (BoW) in these records. The vector has 26435 words, with each word matching a number. This is done with the following code.

```
from sklearn.feature_extraction.text import
CountVectorizer
bow_transformer =
CountVectorizer(analyzer=text_process).fit(X)
len(bow_transformer.vocabulary_)
Output:26435
```

Based on this vector and all 4086 reviews, the Sparse matrix is formed with rows equal to the number of words and columns of the individual reviews (26436x4086), in which the word/review cell records the number of encounters of a particular word in a given review (Fig. 3.).

Reviews Counters	Review 1	Review 2	Review 3	Review ...	Review N
Word 1	0	0	2	1	2
Word 2	1	2	0	2	0
Word 3	0	0	3	0	1
Word ...	1	4	0	0	0
Word M	3	0	2	3	0

Fig. 3. *Sparse Matrix*

Based on this matrix and Multinomial Naïve Bayes classifier, the model is trained to predict the class of a review. For model training, multiple reviews are divided into 80% training reviews and 20% for testing. It can be seen in the following code:

Preparation of training and testing sets:

```
X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.2,
random_state=101)
```

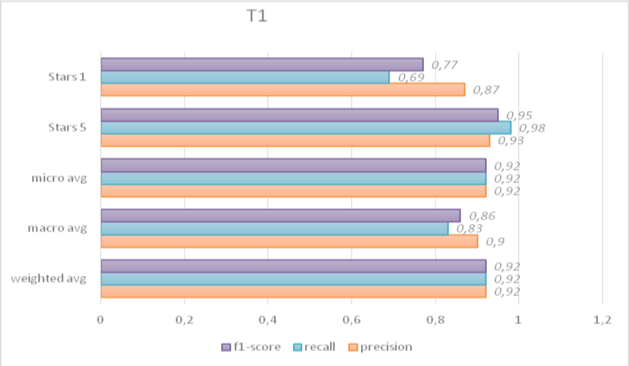
Model training:

```
from sklearn.naive_bayes import MultinomialNB
naive_b = MultinomialNB()
naive_b.fit(X_train, y_train)
```

Accuracy or Precision is a simple model evaluation metric. It is used as the evaluation metric of the different algorithms, it is defined as the percentage of the labels correctly predicted by a model. For example, if a test dataset has 100 observations and a model correctly predicts the labels for 85 observations, its accuracy is 85 %.

As can be seen from the table 1 below, the weighted average Precision, F1-Score and Recall MNB model are 92%.

Table 1. *Multinominal Naïve Bayes prediction precision (Stars1 and 5)*



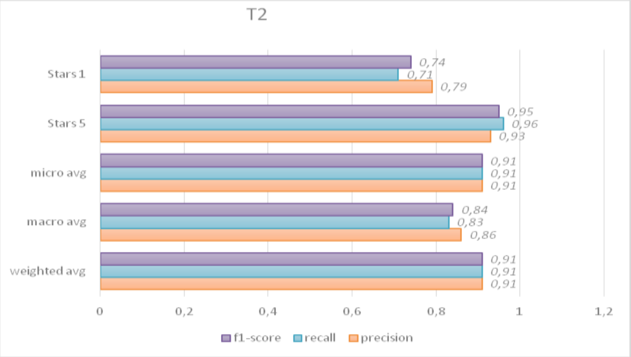
The same steps are implemented for Support Vector Machines classification using Linear classificator.

Model training:

```
from sklearn.svm import SVC
svclassifier = SVC(kernel='linear')
svclassifier.fit(X_train, y_train)
```

The results of Linear SVM precision (91%) are provided in table 2.

Table 1. *Linear SVM prediction precision(Stars1 and 5)*



The precision results of both algorithms are similar 91%-92 % but the multinomial NB sometimes produce wrong prediction on some negative labelled reviews because of the overfitting (example 3). It can be seen in the examples below.

Example 1. Right prediction on positive review

```
positive_review_transformed =
bow_transformer.transform([positive_review])
print("MNB prediction
Stars:",naïve_b.predict(positive_review_transformed
)[0])
print(("SVM prediction
Stars:",svclassifier.predict(positive_review_transf
ormed)[0])
Results:
MNB prediction Stars: 5
SVM prediction Stars: 5
```

Example 2. Right prediction on negative review

```
negative_review_transformed =
bow_transformer.transform([negative_review])
print("MNB prediction
Stars:",naïve_b.predict(negative_review_transformed
)[0])
print("SVM prediction
Stars:",svclassifier.predict(negative_review_transf
ormed)[0])
Results:
MNB prediction Stars: 1
SVM prediction Stars: 1
```

Examle 3. Wrong prediction on negative review of MNB due to overfitting and correct result of SVM

```
next_negative_review_transformed                  =
bow_transformer.transform([next_negative_review])
print(("MNB prediction
Stars:",naïve_b.predict(next_negative_review_transf
ormed)[0])
print ("MNB prediction
Stars:",svclassifier.predict(next_negative_review_t
ransformed)[0])
Results:
MNB prediction Stars: 5   - Overfitting
SVM prediction Stars: 1   - Correct
```

5.1. Algorithms with Star1, 2, 4 5 subset of Yelp

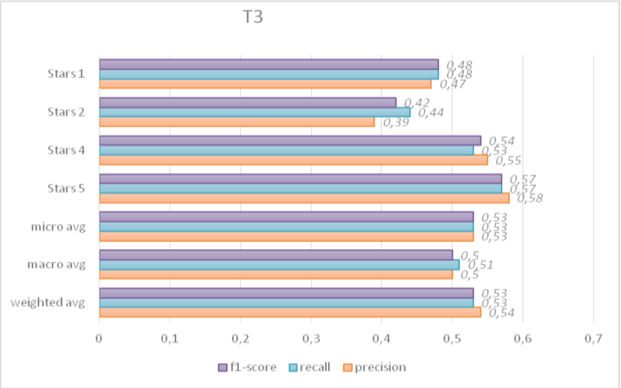
With second two algorithms, based on MNB and SVM, the data was extended by retrieving the Stars 1, 2 and Stars 4, 5 scores that give meaning to a positive (4 and 5) or negative (1 and 2) estimations. Thus, the dataset is extended to 8539 records.

```
yelp_class=yelp[(yelp['stars']==1)|(yelp['stars']
==2)| [(yelp['stars']==4)|(yelp['stars'] ==5)]
yelp_class.shape
Output: (8539, 10)
```

On the new subset the same steps as in the previous algorithms are applied. The only difference is that because of more reviews the BoW dictionary extends to 40526 words, so the Sparse Matrix grows to 40526x8539. That increases the time for its processing.

The Precision results of both Multinomial NB and Linear SVM are provided in tables 3 and 4.

Table 3. *Multinominal Naïve Bayes prediction precision (Stars1,2,4,5)*



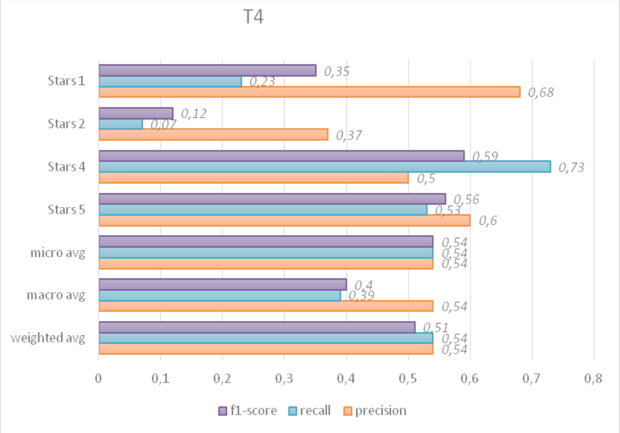
Here because the negative sentiment is related to categories Stars 1 and 2 and positive sentiment to Stars 4 and 5 the Precision results vary between 42% and 53% but prediction is still 92%-92%.

Again the same results related to the overfitting with MNB algorithm are observed, and again the Linear SVM produces correct prediction results in this case (see example 6).

Now because negative sentiment is related to Stars 1 and 2 in the example 5 the review with Star 2 is correctly classified as negative. The same is valid for positive review in example 4 (positive predicted with Star 5).

Because of the fixed volume of the paper the ytexts of the reviews are omitted.

***Table 4**. Linear SVM prediction precision (Stars1,2,4,5)*



Example 4. Right prediction on positive review

```
Results:
MNB prediction Stars: 5
SVM prediction Stars: 5
```

Example 5. Right prediction on negative review

```
Results:
MNB prediction Stars: 2
SVM prediction Stars: 2
```

Example 6. Wrong prediction on negative review of MNB due to overfitting and correct result of SVM

```
Results:
MNB prediction Stars: 4
SVM prediction Stars: 1
```

6. Conclusion

A sentiment analysis is one of the most interesting techniques to find out the users opinion of products. The algorithms and Python programs proposed in this paper are able to perform the text reviews sentiment analysis over the large amount of data with high speed near to real-time. The programs are based on Python libraries NLTL, Scikit-learn and Panda which provide compact code due to the fact that a great deal of data processing is encapsulated in libraries, which saves developers much effort and time.

Implemented Naïve Bayes and SVM algorithms are applied twice on different subsets of Yelp dataset. In first program we compare the precision of both algorithms using subset of Yelp containing review ranked by the customers with stars 1(negative) and 5(positive). In second program the same algorithms are applied on reviews with Stars 1,2 (negative) and Stars 4,5 (positive). The prediction results, as well the precision of MNB and Linear SVM models are similar (91%-93%) but Linear SVM model classifies better the with the overfitting which is a problem for MNB one.

The sentiment analysis programs developed here can be applied in addition to evaluating the quality of services or goods, as well for evaluating the quality of individual lectures or lectures in the university system. The assessments and recommendations formed by these programs can serve as a basis for students to choose the most suitable courses to visit, and lecturers to serve as feedback for updating the lecture material.

7. Acknowledgements

The research is funded by the Research Fund (FNI) of the Ministry of Education, under the project "Synergy between procedural philosophy and elements of artificial intelligence in the theory of education" № DN 15/9 from 11.12.2017.

8. References

1. Hackelingm G.,Mastering Machine Learning with scikit-learn, 2014, Pakt Publishing

2. Lane H., Howard C., Hapke H. M., Natural Language Processing in Action Understanding, analyzing, and generating text with Python, Manning Publications Co., Shelter Island, NY

3. Palash, Goyal, Sumit, Pandey, Karan, Jain, Deep Learning for Natural Language Processing: Creating Neural Networks with Python, ISBN-13 (pbk): 978-1-4842-3684-0

4. Chopra D., Joshi N., Mathur I., Mastering Natural Language Processing with Python, Copyright © 2016 Packt Publishing

5. Ahmad M., Document Classification Using Python and Machine Learning, Digital Vidya, Dec. 2018, https://www.digitalvidya.com/blog/document-classification-python-machine-learning/

6. Ramesh R., Divya G., Divya D., Merin K., Vishnuprabha V., Big Data Sentiment Analysis using Hadoop, IJIRST, Volume 1, Issue 11, pp. 92-98, 2015.

7. Zainuddin N., Selamat A., Sentiment Analysis Using Support Vector Machine, IEEE International Conference on Computer, Communication, and Control Technology (I4CT 2014), Kedah, Malaysia,pp.333-337, 2014.

8. Mehta R., Big Data Analytics with Java, Packt Publishing Ltd, ISBN 978-78728-898-0, UK, 2017.

9. Al-Barznji K., Atanassov., A Framework for Cloud Based Hybrid Recommender System for Big Data Mining, Journal of Science, Engineering & Education, Volume 2, Issue 1, UCTM, Sofia, Bulgaria, pp. 58-65, 2017.

10. Guller M., Big Data Analytics with Spark, ISBN-13 (pbk): 978-1-4842-0965-3, 2015.

11. Pentreath N., Machine Learning with Spark, Packt Publishing Ltd. Birmingham – Mumbai, 2015.

12. https://spark.apache.org/docs/latest/, Online Feb 2018.

13. Fang X., Zhan J., Sentiment analysis using product review data, Joournal of Big Data, pp. 1–14, 2015.

14. 7. Atanassov A., Al-Barznji K., Tomova F., System for Sentiment Analysis of Big Text Data, International virtual journal for science, techniques and innovation for the industry MTM, Issue 8, /2018 ISSN 1313-0226