SYSTEM FOR SENTIMENT ANALYSIS OF BIG TEXT DATA

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

Abstract: *The importance of Big Data and Big Data Mining is growing significantly in recent years. Different kind of e-sources as social networks, e-commerce sites, e-mails, sensors, etc. are generating large amount of structured and unstructured numerical and text data. This data provides valuable information about costumer's preferences or ratings of products or commodities. This information is essential for making predictions on the base of the sentiment analysis of this data. The sentiment analysis of large amount of text data requires specific big data and machine learning /ML/ libraries. In this paper the implementation of a system for big data sentiment analysis using ML algorithms is proposed. It is based on Naïve Bayes and Support Vector Machines /SVM/ classification ML algorithms for text analysis. The system is implemented in Java and uses Apache Spark ML libraries which are very flexible, fast and scalable. The system is tested with well known Amazon dataset and its performance is measured in form of accuracy. The obtained results approve the effectiveness of big data sentiment analysis algorithms. The System can be applied for recommendation of products and services or predictions of customers' needs*.

Keywords: BIG DATA; MACHINE LEARNING; SENTIMENT ANALYSIS; NAÏVE BAES; SVM

1. Introduction

Big Data is a new term used to identify the data sets that are of large size and have greater complexity [1]. Big data is defined as a large amount of data which requires new technologies and architectures to make possible to extract value from it by capturing and analysis process [2]. The Big data is very important for business and society purposes. The data came from different sources like sensors used to accumulate climate information, available data on the social media websites, video movie audio and so on. This series of data are known as Big Data [3]. They are in form of structured, semi-structured or unstructured data and can be used for Sentiment Analysis [4].

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].

Machine learning /ML is a branch of artificial intelligence [8]. ML is the science of training a system to learn from data and act. [10]. ML computations aim to derive predictive models from current and historical data. The inherent premise is that a learned algorithm will improve with more training or experience, and in particular, ML algorithms can achieve extremely effective results for specific domains using models trained from large datasets [11]. Machine learning is used for a variety of tasks in different fields, as well for Sentiment Analysis. The ML tasks can be broadly grouped into Classification, Regression, Clustering, Anomaly detection, Recommendation, and Dimensionality reduction [10].

2. Machine Learning Algorithms

Machine learning algorithms are generally grouped into two categories (Fig. 1): supervised and unsupervised learning [9].

2.1. Supervised Machine Learning Algorithms

Supervised machine learning refers to working with a set of labelled training data to learn [8], which means, they can be used only with labelled training datasets. Each observation in the training dataset has a set of features and a label. A supervised machine learning algorithm learns from data to estimate or approximate the relationship between a response variable (label) and one or more predictor variables (features). The labels in a training dataset may be generated manually or provided from another system [10]. Supervised machine learning algorithms can be grouped into Recommendation Engines, Regression and Classification algorithms [12]. The commonly used supervised algorithms for regression tasks are Linear Regression, Decision Trees, and Ensembles of Trees. The classification tasks are related to Logistic Regression, Support Vector Machines /SVM/, Naïve Bayes, and different types of Artificial Neural Networks. The Recommendation includes Collaborative filtering with Alternating Least Squares [10].

Machine Learning Algorithms	
Supervised Machine Learning	**Unsupervised Machine Learning**
Regression	**Clustering**
Classification	**Anomaly detection**
Recommendation	**Dimensionality reduction**

Fig. 1: *Machine learning algorithms*

2.2. Unsupervised Machine Learning Algorithms

Unsupervised machine learning is related to cases when the model does not require labelled data. These types of models try to learn or discovering hidden structures in unlabeled data or reduce the data down to its most important features [12]. With unsupervised learning there is no right or wrong answer [8]. They are generally used for clustering, anomaly detection, and dimensionality reduction. The list of commonly used unsupervised algorithms includes k-means, Principal Component Analysis, and Singular Value Decomposition.

2.3. Sentiment Analysis

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 [14] 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. ML is one of the widely used approaches towards sentiment classification. Sentiment analysis has been applied to the broader area of research including consumer product reviews and services [5]. One of the forms of text analysis is SA. So to understand what kind of emotion (love, hate, positive, negative, and so on) contains certain text, the techniques of SA are requred [6].

2.4. Natural Language Processing (NLP)

Some techniques and concepts from NLP will be used working on sentimental analysis problems. Some of them are given below:

- ***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 [15].

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].

2.5. 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 [16]. 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 [6].

2.6. 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 Big Data Machine Learning Libraries*

From the machine learning point of view, there is a fundamental difference in algorithm implementation to process the data in memory or from distributed storage. In case the processed data cannot be stored in the memory, then a big data machine learning library [9] is required. There are a lot of Big Data Machine Learning Libraries developed on programming languages as Python, C#, C++, but here the focus is on the Apache Spark libraries for Java, because the proposed in the paper system is developed using Java.

3.1 Apache Spark

Apache Spark is a fast and general engine for large-scale data processing. Spark supports main-memory caching and possesses a loop-aware scheduler. Spark is fully compatible with Java-based applications. Its own programming language Scala fully supports a Java Virtual Machine and allows easier integration with Java [13]. Spark provides two machine learning libraries: - MLlib and Spark ML (also known as the Pipelines API). Spark MLlib is Spark's implementation of the machine learning algorithms based on the RDD format. Resilient Distributed Datasets (RDDs) is a distributed main-memory abstraction that enables users to perform in-memory computations on large systems. MLlib is the first machine learning library that is shipped with Spark. It is much faster and scalable than single node machine learning libraries. Spark ML is based on datasets and allows usage of Spark SQL along with them. Feature extraction and feature manipulation tasks become very easy because they can be handled using Spark SQL queries. Spark ML is provided with an advanced feature - Pipeline. Plain data is usually in an extremely raw format and this data goes through a cycle or workflow where it gets cleaned, mutated, and transformed before it is used for processing and training of machine learning models. This entire workflow of data and its stages are very well encapsulated in the new Pipeline of the Spark ML library [6]. In addition, since Spark allows an application to cache a dataset in memory, machine learning applications built with Spark ML or MLlib are fast [10]. Spark ML help users to create and tune practical machine learning pipelines [11]. Spark ML libraries contain implementations of many algorithms and utilities that can be used for common machine learning tasks such as Regression and Classification, Clustering, Dimensionality Reduction, Feature Extraction and Transformation, Frequent pattern mining, and Recommendation [10]. Spark ML supports the implementation of multinomial Naïve Bayesian and Bernoulli Naïve Bayes, as well linear SVM algorithms [13].

3.2 ML Pipelines

In machine learning, it is common to run a sequence of algorithms to process and learn from data. For example, a simple text document processing workflow might include several stages as splitting document's text into words, converting each document's words into a numerical feature vector; learning a prediction model using the feature vectors and labels. MLlib represents such a workflow as a Pipeline, which consists of a sequence of Pipeline Stages (Transformers and Estimators) to make it easier to combine multiple algorithms into a single pipeline, or workflow, to be run in a specific order. ML Pipelines provide a uniform set of high-level APIs built on top of Data Frames that help users create and tune practical machine learning pipeline [13].

4. System for Sentiment Analysis

The developed System for Big Data Sentiment Analysis, object of this paper, is based on machine learning classification algorithms for text analysis. The goal of the system is to train a predictive model that predicts whether a sentence has a positive or negative sentiment. As given in Fig. 2 the system takes Big Data sets as input, and then performs the sentiment analysis for selected data set by using provided by the Spark ML library Naïve Bayes and SVM classification algorithms for text analysis.

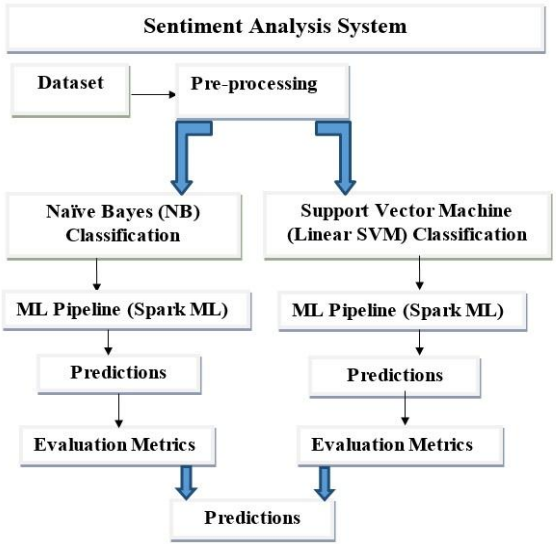


Fig. 2: *Sentiment Analysis Algorithms for analyzing big data and generating predictions*

The results were measured using Accuracy, for evaluating the effectiveness of the proposed methods. The system is implemented

using Java programming language and mentioned above Spark ML library. It is trained and evaluates multiclass classifiers using the labelled dataset from the amazon_cell_labelled.txt.

The algorytm of developed system is described in the pseudo code below. It includes following steps:

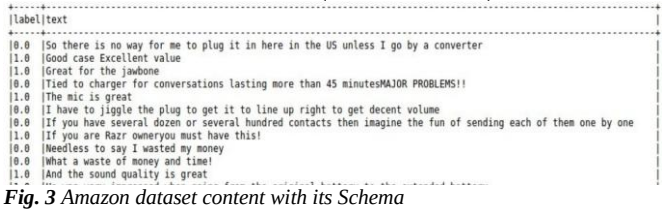
First (step 1), the dataset is loaded as input to the system and (step 2) the data sparse is performed. After caching (step 3 and 4)the dataset into the system, before applying and evaluating models the dataset is split into 80 % train dataset and 20 % test dataset randomly (step 5 and 6). Then the pre-processed steps (step 7, 8 and 9) will start on the training dataset using some concepts of NLP for sentiment analysis such as Feature Extractors and Feature Transformers.

Steps 10 to 21 are the same and are applied in parallel for both Naïve Bayes and Linear SVM models. The parallel implementation is performed using the Spark ML Pipeline (steps 11 and 12) for training both models. Getting the predictions (step 13) and estimation of the average accuracy (steps 14 to 20) and test error (step 21) is done in the remaining steps. Finally, in step 22 the total prediction is produced.

procedure Naïve Bayes and Linear SVM using Spark ML
Input: (Labelled-dataset.txt) [sentence, label], From local file system.
Output: label, text, words, updateWords, rawFeature, features, probability, prediction.
1. Load the data from Labelled-dataset.txt into an RDD.
dataset1←(labelled-dataset.txt).
2. Sparse the data for that: Split the data based on the tab (‘\t’) and return
reviewsRdd ← dataset1.map(strRow → [] rowArr
←strRow.split("\t"),
rawText ← rowArr[0]
realText ← rawText.replaceAll(",", "").replaceAll("\"", "")
.replaceAll("*", "").replaceAll("\\.", "").trim();
return (setText(realText) and
setLabel(Double.parseDouble(rowArr[1])))
3. Then returns Dataset of Review class objects productsDs ←
spark.createDataFrame(reviewsRdd.rdd(), Review.class);
4. Store the productsDs data in memory using cache() , or
productsDs.createOrReplaceTempView("Review");
5. compute K iteration for get average performance evaluation accuracy
based on K-fold Cross-validation idea:
k=5; double accuracyArr[] =new double[k];
for (i =1 to k) (repeat k iteration)
6. randomSplit() the Dataset (productsDs) into trainingDataset (80%) and
tesDataset (20%).
7. perform Tokenizer to convert texts to words tokenizer ←
newTokenizer().setInputCol("text").setOutputCol("words");
8. Perform StopWordsRemover to remove stop words
stopWordRem←newStopWordsRemover()
.setInputCol("words") .setOutputCol("updatedWords");
9. perform TF-IDF and numFeatures ← 20000;
hashingTF ← newHashingTF().setInputCol("updatedWords")
setOutputCol("rawFeatures").setNumFeatures(numFeatures);
idf ←newIDF().setInputCol("rawFeatures")
.setOutputCol("features");
10. perform NaiveBayes / LinearSVM algorithms
NB ← new NaiveBayes().setFeaturesCol("features")
.setPredictionCol("predictions");
LinearSVM← newLinearSVC().setFeaturesCol("features")
.setPredictionCol("predictions");
11. perform Pipeline for combining all methods together
pipe ← pipe.setStages(new PipelineStage[]{ tokenizer, stopWrdRem,
hashingTF, idf, NB/ LinearSVM });
12. Fit the pipeline to training data using PipelineModel
pipemodel ← pipe.fit(trainingDs);
13. getting predictions via testdataset, predicts←
model.trasform(testDataset)
14. perform Multiclass Classification Evaluator
evaluator ← calculate (Accuracy metrics) on predicts
15. accuracyArr[i]← accuracy; // for storing all k accuracy
16. end for
17. define and declaration double SumAccuracy=0 , AverageAccuracy=0;
18. for (m=1 to accuracyArr.length)
SumAccuracy← SumAccuracy + accuracyArr[m];
19. end for
20. return AverageAccuracy← SumAccuracy/accuracyArr.length;
21. return Test Error ← (1.0 - AverageAccuracy));
22. return predicts.show(false) to show all fields from predict dataset (label, text, words, updateWords, rawFeature, features, probability, prediction).
end procedure

Dataset used for evaluating the system

The sentiments labelled sentences datasets are available at https://archive.ics.uci.edu/ml/datasets/. For current system tests the Amazon cell phone reviews dataset (amazon_cell_labelled.txt) is used. It includes randomly selected 500 positives and 500 negatives reviews. A review with negative sentiment is labelled **0** and with positive review is labelled **1**. A review is separated from its label by the tab character. On Fig. 3**,** is given the sample of the content of Amazon dataset with its schema (label/text review).



label	text
0.0	So there is no way for me to plug it in here in the US unless I go by a converter
1.0	Good case Excellent value
1.0	Great for the jawbone
0.0	Tied to charger for conversations lasting more than 45 minutesMAJOR PROBLEMS!!
1.0	The mic is great
0.0	I have to jiggle the plug to get it to line up right to get decent volume
0.0	If you have several dozen or several hundred contacts then imagine the fun of sending each of them one by one
1.0	If you are Razr owneryou must have this!
0.0	Needless to say I wasted my money
0.0	What a waste of money and time!
1.0	And the sound quality is great

Fig. 3 *Amazon dataset content with its Schema*

Train and Test Datasets

In order to estimate the generalization error the input data set is spitted into two datasets randomly: training dataset and testing dataset [9]. First one contains 80% from the original dataset and second one 20%. First one is used of building models, which is used for training the models to get predictions / recommendations. The second one is used for testing and finding the performance evaluating metrics such as accuracy and error and for evaluating the quality of the training models.

Cross-validation

Cross validation splits the input dataset into k sets of almost the same size, for example, to five sets as shown in Fig 4. First, sets 2-5 will be used for training and set 1 for testing. Then the procedure is repeated five times, leaving out one set at a time for testing, and average the error over the five repetitions [9].

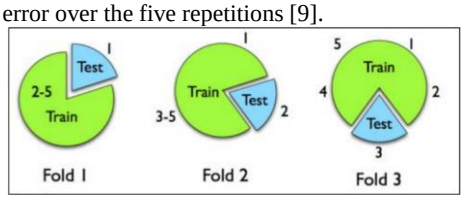


Fig. 4 *Cross-validation K- Fold*

Pre-processed datasets

The pre-processing or cleaning the datasets is performed via NLP concepts (Features Transformers and Extractors), to convert an input string to lowercase and to split it into words (tokens) using whitespaces as a separator, and eliminate not need tokens.

label	text	words	updatedWords	rawFeatures	features
0.0	-Drinks took clos...	[-drinks, took, c...	[-drinks, took, c...	(20000,[419,2044,...	(20000,[419,2044,...
0.0	2 times - Very Ba...	[2, times, -, ver...	[2, times, -, bad...	(20000,[1413,5499...	(20000,[1413,5499...
0.0	Although I very m...	[although, i, ver...	[although, much, ...	(20000,[2745,7190...	(20000,[2745,7190...
0.0	And it was way to...	[and, it, was, wa...	[way, expensive]	(20000,[5062,1915...	(20000,[5062,1915...
0.0	Any grandmother c...	[any, grandmother...	[grandmother, mak...	(20000,[941,2044,...	(20000,[941,2044,...
0.0	As much as I'd li...	[as, much, as, i'...	[much, like, go, ...	(20000,[3330,8430...	(20000,[3330,8430...
0.0	But I don't like it	[but, i, don't, l...	[like]	(20000,[3330],[1.0])	(20000,[3330],[3....
0.0	Coming here is li...	[coming, here, is...	[coming, like, ex...	(20000,[3330,7909...	(20000,[3330,7909...
0.0	Del Taco is prett...	[del, taco, is, p...	[del, taco, prett...	(20000,[2061,3075...	(20000,[2061,3075...
0.0	Disappointing exp...	[disappointing, e...	[disappointing, e...	(20000,[2745,7190...	(20000,[2745,7190...
0.0	Don't waste your ...	[don't, waste, yo...	[waste, time]	(20000,[15866,181...	(20000,[15866,181...
0.0	Food is way overp...	[food, is, way, o...	[food, way, overp...	(20000,[1742,8653...	(20000,[1742,8653...
0.0	Gave up trying to...	[gave, up, trying...	[gave, trying, ea...	(20000,[213,1800,...	(20000,[213,1800,...
0.0	He was extremely ...	[he, was, extreme...	[extremely, rude,...	(20000,[8463,1131...	(20000,[8463,1131...
0.0	Hell no will I go...	[hell, no, will, ...	[hell, go, back]	(20000,[1617,8430...	(20000,[1617,8430...

Fig. 5 *Pre-processed steps results for Amazon Dataset*

To prepare data in a format that can be used in machine learning algorithms a feature Vector for each sentence in the dataset must be created. For that the TF-IDF is applied to get a bag of words. Then the Sentimental analysis algorithms can be applied using a bag of words. Figure 5 shows a sample of the results of the pre-processed steps of the selected dataset.

Results

The results of both machine learning algorithms for text analysis on the Amazon_cell_labelled dataset are shown in Fig. 6 and Fig.7. These results are the final results of testing the training models by test dataset, after performing the training models on train dataset. The Naïve Bayes model produces the probabilities and predictions from the features, but the Linear SVM model produces

only the predictions and their results easily can be compared with the base value of labels.

label	features	rawPrediction	probability	predictions
0.0	(20000,[19637],[5...]	[-40.198273587791...]	[0.99999962538923...]	0.0
0.0	(20000,[941,2745,...]	[-210.19583354933...]	[0.99999981652783...]	0.0
0.0	(20000,[2044,3208...]	[-554.98779945383...]	[0.99996119971449...]	0.0
0.0	(20000,[1624,4034...]	[-332.79628117605...]	[0.99999997767548...]	0.0
0.0	(20000,[505,2277,...]	[-128.90480026635...]	[0.99999999893763...]	0.0
0.0	(20000,[1955,2177...]	[-347.08097989511...]	[0.01932759708368...]	1.0
0.0	(20000,[1868,1924...]	[-301.97116641090...]	[0.00678859272681...]	1.0
0.0	(20000,[4511,8103...]	[-112.75167131188...]	[0.99999999999998...]	0.0
0.0	(20000,[17086,192...]	[-58.228223650788...]	[0.99999817092654...]	0.0
0.0	(20000,[2404,1347...]	[-52.797394111466...]	[0.02759563603203...]	1.0
0.0	(20000,[8297,1238...]	[-68.913703017755...]	[0.97675789312516...]	0.0
0.0	(20000,[16213],[4...]	[-28.086391713991...]	[0.85120876645934...]	0.0
0.0	(20000,[15984,162...]	[-47.195515557507...]	[0.65350430047939...]	0.0
0.0	(20000,[1413,5499...]	[-105.49344847043...]	[0.99818726296542...]	0.0
0.0	(20000,[13337,135...]	[-72.095790908299...]	[0.99999999976954...]	0.0

Fig. 6 *Sentiment Analysis Results by using Naïve Bayes*

label	features	rawPrediction	predictions
0.0	(20000,[645,3866,...]	[-0.5475272933290...]	1.0
0.0	(20000,[941,2745,...]	[0.15520851708257...]	0.0
0.0	(20000,[2044,3208...]	[1.53501873581684...]	0.0
0.0	(20000,[8418,9694...]	[-0.4611041347706...]	1.0
0.0	(20000,[15984,190...]	[-1.1296972759785...]	1.0
0.0	(20000,[4511,8103...]	[1.09560932818885...]	0.0
0.0	(20000,[4366,1159...]	[0.97878584992358...]	0.0

Fig. 7 *Sentiment Analysis Results by using Linear SVM*

Model Evaluations

Accuracy 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 test dataset contains 100 observations and a model correctly predicts the labels for 85 observations, then its accuracy is 85 %. To evaluate how the generated model performs on both the training and test datasets, first the predictions is get for each observation in the training and test datasets. Second, the models are evaluated by using accuracy measure to the test datasets. To improve the accuracy evaluation performance results are used in the K-fold cross-validation idea with k=5 iterations.

Table 1: *Techniques Evaluation Metrics Results*

Techniques with Evaluation Metrics	**Naïve Bayes**	**Linear SVM**
Accuracy Evaluation Metrics1	0.8	0.79803
Accuracy Evaluation Metrics2	0.78421	0.7973
Accuracy Evaluation Metrics3	0.77941	0.83333
Accuracy Evaluation Metrics4	0.77251	0.80729
Accuracy Evaluation Metrics5	0.80303	0.8
Average Accuracy	**0.78783**	**0.80719**
Test Error	**0.21217**	**0.19281**

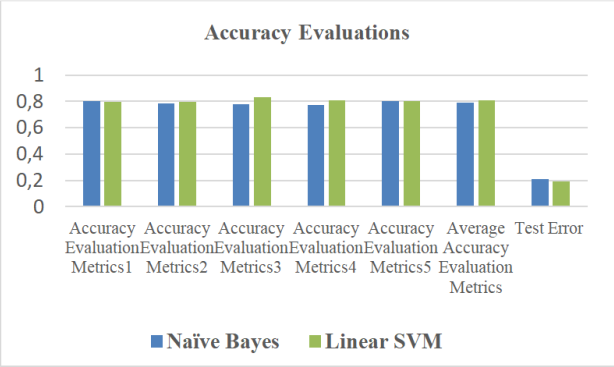


Fig. 8 *Evaluation Metrics Results*

The results are presented in Table 1 and Fig. 8. The Average accuracy evaluation metrics detects that the Linear SVM model is better than Naïve Bayes model using the test datasets to evaluate of this dataset. Because the accuracy score will be between 0 and 1, and high Accuracy score is better, it means that has less error.

5. Conclusion

Big data sentiment analysis is one of the most interesting techniques to find out the users opinion of products. The system proposed in this paper is able to perform the text reviews sentiment analysis over the large amount of data with high speed near to real-time. The system uses Apache Spark ML libraries to implement big data sentiment processing on the base of Naïve Bayes and SVM models which classify the reviews in positive and negative sentiments. The work of proposed models implements several stages: pre-processing stage, features generation stage, classifiers learning stage, and Pipeline stage. The analytical evaluation of proposed model is done by accuracy evaluation measure. It is shown through experiments that the performance of the system is increased by obtaining the average evaluation accuracy metrics, based on the cross-validation idea. The comparison analysis in the proposed system shows that Linear SVM is better than Naïve Bayes in term of average accuracy metrics. The main focus of current research was to perform the Sentiment Analysis faster and Big Data sets to be handled efficiently. At the moment the system is implemented on a single node PC configuration. Next steps in the future will be to run the system in a multi-node configuration and to increase its performance, as well to test it with larger data sets.

5. References

1 Bifet A., Mining big data in real time, Conference on 100 years of Alan Turing and 20 years of SLAIS 11,2012, Ljubljana, Slovenia

2 Venkata S. L., A Survey on Challenges and Advantages in Big Data, www.ijcst.com/vol62/1/24-lenka-venkata-satyana-rayana.pdf, pp. 115–119, 2015.

3. Verma J. P., Smita A., Patel B., Atul P., Big Data Analytics: Challenges and Applications for Text, Audio, Video, and Social Media Data, Int. J. Soft Comput. Artif. Intell. Appl., vol. 5, no. 1, pp. 41–51, 2016.

4. 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.

5. 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.

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

7. 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.

8. Bell J., Machine Learning: Hands-On for Developers and Technical Professionals, John Wiley & Sons, Inc., Indiana, 2015.

9. Kaluža B., Machine Learning in Java, Packt Publishing Ltd, UK, 2016.

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

11. Bengfort B., Kim J., Data Analytics with Hadoop, Published by O'Reilly Media, Inc., First Edition. USA, 2016.

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

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

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

15. https://machinelearningmastery.com/ Online Feb 2018.

16. Baltas A., Kanavos A., Tsakalidis A. K., An Apache Spark Implementation for Sentiment Analysis on Twitter Data, Patras, Greece, Springer International Publishing AG, pp. 15-25, 2017.