Sentiment Analysis using supervised classification algorithms

Yassine AL-AMRANI
LIROSA Laboratory
National School of Applied Sciences
Tetuan, Morocco
alamraniyassine@gmail.com

Mohamed LAZAAR
New Technology Trends Team
National School of Applied Sciences
Tetuan, Morocco
lazaarmd@gmail.com

Kamal Eddine ELKADIRI
LIROSA Laboratory
National School of Applied Sciences
Tetuan, Morocco
elkadiri@uae.ma

Abstract — The exploitation of social media (forums, blogs and social networks) has become crucial due to the explosive growth of textual data from these new sources of information. Our work focuses on the Sentiment analysis resulting from the messages (SMS, Facebook, Twitter...) using original techniques of search of texts. These messages can be classified as having a positive or negative feeling based on certain aspects in relation to a query based on terms. This paper presents a comparison of five supervised classification algorithms: PART, Support Vector Machine, Decision Tree, Naive Bayes, and Logistic Regression.

Keywords
Sentiment analysis, social media, Support vector machines, Decision tree, Naive Bayes, Logistic Regression, PART, classification.

BDCA'17, March 29-30, 2017, Tetouan, Morocco

ACM ISBN 978-1-4503-4852-2/17/03…$15.00
http://dx.doi.org/10.1145/3090354.3090417

1. INTRODUCTION

Lately, new media such as social networks (Facebook, Twitter, LinkedIn ...) are developing very interesting either in terms of volume of data or according to the number of users around the world. They offer users all the possibilities to express their opinions and to exchange their ideas with the others through multiple platforms like the SMS and the emails [1].

Sentiment analysis is the part of the text mining that attempts to define the opinions, feelings and attitudes present in a text or a set of text. It is particularly used in marketing to analyze for example the comments of the Net surfers or the comparatives and tests of the bloggers or even the social networks. Sentiment analysis requires much more understanding of the language than text analysis and subject classification. Indeed, if the simplest algorithms consider only the statistics of frequency of occurrence of the words, it is usually insufficient to define the dominant opinion in a document, especially when the content is short as messages. It is the process of determining the contextual polarity of the text, that is, whether a text is positive or negative. The use of this analysis helps researchers and decision-makers better understand opinions and client satisfaction using sentiment classification techniques in order to automatically collect different perspectives on from various platforms. There has been a large amount of research in the area of sentiment classification. Traditionally most of it has focused on classifying larger pieces of text, like reviews [2].

In this paper, a comparison of five popular classifiers was performed to classify SMS text either positive or negative (Decision Tree (DT), Support Vector Machine (SVM), Naive Bayes (NB), PART and Logistic Regression (LR)).

2. SENTIMENT ANALYSIS SYSTEM

Sentiment analysis is the field of study that analyzes people's opinions, sentiments toward entities such as products, services, etc... [3]. A probabilistic approach for SMS classification systems has been proposed by [4]. Recently, sentiment analysis has attracted an increasing interest. It is a hard challenge for language technologies, and achieving good results is much more difficult than some people think. The task of automatically classifying a text written in a natural language into a positive or negative feeling, opinion or subjectivity (Pang and Lee, 2008), is sometimes so complicated that even different human annotators disagree on the classification to be assigned to a given text. Personal interpretation by an individual is different from others, and this is also affected by cultural factors and each person's experience. And the shorter the text, and the worse written, the more difficult the task becomes, as in the case of messages on social networks.

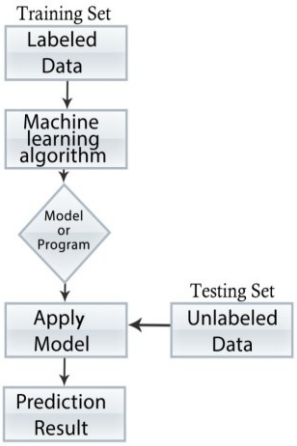


Fig. 1 Supervised Machine learning system approach for Sentiment Analysis

In feature extraction, a sentence or document is broken into words to build up the feature matrix. In the matrix, each sentence or document is a row and each word form a feature as a column, and the value is the frequency count of the word in the sentence or document. Feature matrix is then passed to each classifier and their performance is evaluated [1].

In this work, we have studied the classification of sentiment using five popular machine learning algorithms, namely Decision Tree, Support Vector Machine, Naive Bayes, Logistic Regression and PART.

Decision Tree classifies data into different classes by recursively separating the feature space into two parts and assigning different classes based upon which region in the divided space a sentence is, based on its features. The Support Vector Machine method is a statistical classification approach which is based on the maximization of the margin between the instances and the separation hyper-plane. The Support Vector Machine (SVM) method was considered the best text classification method [5]. The Naïve Bayes method has been a very popular method in text categorization because its simplicity and efficiency [6]. Logistic regression is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes). PART is a separate-and-conquer rule learner proposed by Eibe and Witten [7]. The algorithm producing sets of rules called decision lists which are ordered set of rules. A new data is compared to each rule in the list in turn, and the item is assigned the category of the first matching rule (a default is applied if no rule successfully matches) [8].

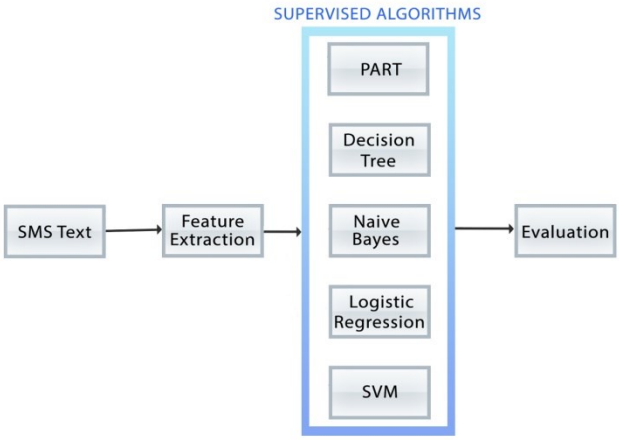


Fig. 2 Control flow of the System

Text classification play an important role in many applications, it assigns one or more classes to a document according to their content. Classes are selected from a previously established taxonomy (a hierarchy of categories or classes). The Text Classification API takes care of all preprocessing tasks required for automatic classification [9] [10].

This API supports a variety of text classification scenarios like:

- Binary classification like spam filtering (HAM, SPAM) or simple sentiment analysis (POSITIVE, NEGATIVE)
- Multiple class classification like selecting one category among several alternatives.

Most partitioning algorithms do not take raw text as input but numeric vectors. For this it is necessary to find a representative transformation that converts the text of the

tweets to digital vectors. A family of this transformation is called Bag-of-Words (BOW).

3. APPLIED ALGORITHMS

In this work, we will apply five supervised learning algorithms

Table. 1 Distribution of selected algorithms and their domains

Group	Algorithms	Areas
Rules-based classifiers	PART	Classification
Classifiers based on decision trees	Decision Tree	
Bayesian networks	Naïve Bayes	
Classifiers 'function'	Logistic Regression	
	Support Vector Machine	

a. RULES-BASED CLASSIFIERS

In this group we treat the PART algorithm:

PART algorithm

PART is a separate-and-conquer rule learner proposed by Eibe and Witten [7]. The algorithm producing sets of rules called decision lists which are ordered set of rules. A new data is compared to each rule in the list in turn, and the item is assigned the category of the first matching rule. PART builds a partial C4.5 decision tree in its each iteration and makes the best leaf into a rule. The algorithm is a combination of C4.5 and RIPPER rule learning [11]. PART (Frank, 1998) makes it possible to infer rules by the iterative generation of partial decision trees by combining two major paradigms: decision trees and the "divide and conquer" rule learning technique. It does not need to perform a global optimization to produce precise sets of rules, which brings more simplicity. It adopts the "divide and conquer" strategy because it builds a rule, removes the instances covered by this rule, and continues to create recursive rules for the remaining instances until none are left.

b. CLASSIFIERS BASED ON DECISION TREES

A decision tree is a tree in which each node represents a choice between a number of alternative solutions, and each leaf node represents a classification or decision.
In this group we treat the Decision Tree algorithm:

Decision Tree (DT)

The first decision tree classification algorithms are old. The two most important works were the creation of CART, by Breiman in 1984 and the creation of C4.5 by Quinlan in 1993. Decision trees are extremely intuitive and provide a graphic, speaking and easy to read representation of an individual's classification protocol. This graphical representation is in the form of a tree consisting of terminal sheets (the classes of individuals) obtained by following a path along the nodes, each node corresponding to a binary question using a variable of the dataset. It's called a decision tree because it starts with a single box (or root), which then branches off into a number of solutions, just like a tree [12].
BAYESIAN NETWORKS
The set of algorithms of this group is based on the Bayes law. Bayes (Cornuéjols, 2002) proposes, on the one hand, that knowledge about the world be translated into a set of hypotheses (finite or not), each of which is affected by a probability reflecting the degree of belief of the learner in the hypothesis in question. Learning then consists in finding a dependency structure between variables or estimating the conditional probabilities defining these dependencies. Thus, models based on this formula allow us to express probabilistic relations between together of facts. The reasoning adopted in this case is conditional.
To build a decision tree, we need to calculate Entropy using the frequency table of one attribute:

$$E(S) = \sum_{i=1}^{C} -P_i \, log_2 \, P_i$$

S : spam or ham
C : number of classes
P : number of messages in class i

c. BAYESIAN NETWORKS

In this group we treat the Naïve Bayes algorithm:

Naïve Bayes (NB)

The Bayesian naive classification is a Bayesian type of simple probabilistic classification based on the Bayes theorem with a strong (or naive) independence of hypotheses. It uses a Bayesian naive classifier or Bayes naive classifier belonging to the family of linear classifiers.
The Naive Bayes algorithm is an intuitive method; it is a simplest model that uses the probabilities of each attribute belonging to each class to make a prediction. This model works well for the categorization of the text. Naive Bayes classifiers assume that the effect of a variable value on a given class is independent of the values of other variable. This assumption is called class conditional independence [13]. It is classification algorithm which makes decision for unknown data set. It is based on Bayes Theorem which describe the probability of event based on its prior knowledge.

The naive Bayesian classifier provides a simple approach with clear semantics to represent, use and learn probabilistic knowledge. This method is used as part of supervised learning. The performance is to accurately predict the class of test instances.

Bayes' theorem is stated mathematically as the following equation:

$$P(H|E) = \frac{P(H) * P(E|H)}{P(E)}$$

where H and E are events and P(E) ≠ 0.

$\boldsymbol{P(H|E)}$, a conditional probability, is the probability of observing event A given that B is true.

$\boldsymbol{P(H)}$ and $\boldsymbol{P(E)}$ are the probabilities of observing A and B without regard to each other.

$\boldsymbol{P(E|H)}$: is the probability of observing event B given that A is true.

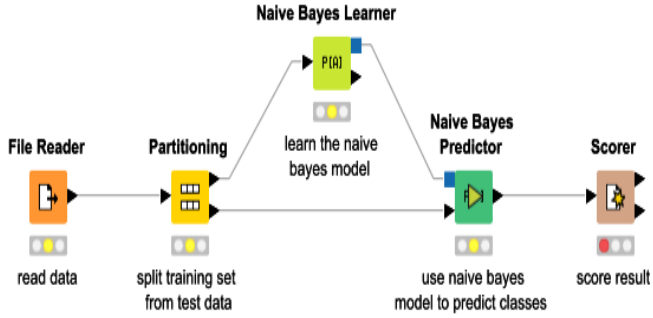


Fig. 3 Diagram shows how naïve bayes works

d. CLASSIFIERS 'FUNCTION'

Several variants exist among other things the Logistic Regression and Support Vector Machine algorithm:

Logistic Regression (LR)

Logistic regression is one of the most popular machine learning algorithms for binary classification. This is because it is a simple algorithm that performs very well on a wide range of problems [14]. Logistic regression corresponds to a linear regression where the dependent variable (or to explain) is binary (ie it can only take two values 0/1 or Yes / No) (Alpaydin, 2004). It is very useful for understanding or predicting the effect of one or more variables on a binary response variable.

The probability for class j with the exception of the last class is:

$$P_j(X_i) = \frac{e^{X_i B_j}}{\left(\sum_{j=1}^{k-1} e^{X_i * B_j}\right) + 1}$$

B : parameter matrix.

k : number of classes.

The last class has probability:

$$1 - \sum_{j=1}^{k-1} P_j(X_i) = \frac{1}{\left(\sum_{j=1}^{k-1} e^{(X_i * B_j)}\right) + 1}$$

Support Vector Machine (SVM)

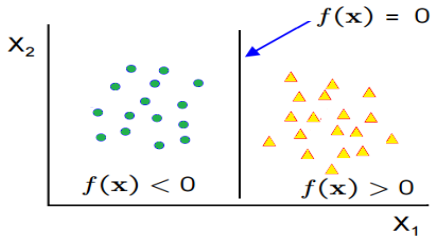
This method was used for the first time in the text classification by [15]. It has proved successful in classifying opinion documents, including style (Diederich et al., 2000). The principle for support vector machine algorithm is to find a classifier, or a discrimination function, whose capacity of generalization (quality of forecasting) is the greatest possible. The examples are represented by points in a space and we look for a (hyper) plane separating at best the classes and that all the observations are the furthest from this plane [16]. The origin of the SVM algorithms is found in the methods developed in the 1960s. The principle is the separation of the learning space by a hyper plane (also called a linear surface) based on the assumption that the set d Learning consists of examples and counter examples. The SVM methods allow taking into account the problem of linearity. They first apply a mathematical transformation to the learning space using kernel functions (these functions are non-linear). Once the transformation is done, the instances can be separated linearly; it remains to find the optimal hyper plane.

To make our discussion of SVMs easier we will be considering a linear classifier for a binary classification problem with labels y and features x. We'll use y ∈ {−1, 1} to denote the class labels and parameters w, b:

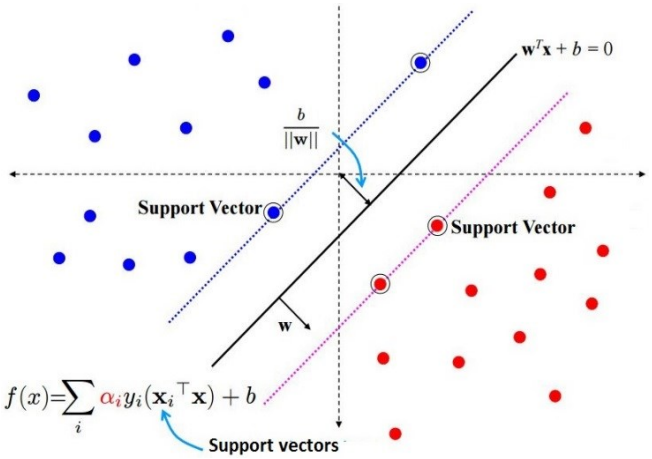
$$\mathbf{f(x) = w^T x + b}$$

w : normal to the line.

b : bias.



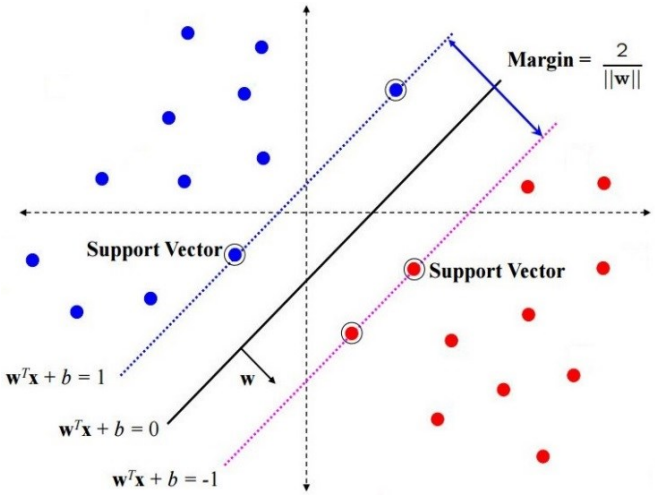
Choose normalization such that ($w^T x_+$ + b = +1) and ($w^T x_-$ + b = -1) for the positive and negative support vectors respectively.



Then the margin is given by:

$$\frac{w}{\|w\|} \cdot (x_+ - x_-) = \frac{w^T(x_+ - x_-)}{\|w\|}$$

$$= \frac{w^T\left(\left(\frac{+1-b}{w^T}\right) - \left(\frac{-1-b}{w^T}\right)\right)}{\|w\|} = \frac{2}{\|w\|}$$



4. RESULTS AND DISCUSSIONS

The training and test data we used for this work were taken from the "SMS Spam Collection Data Set" which contains 5574 SMS divided into two types: positive and negative. But we just worked with the first 200 messages that consist of 167 positive ("ham") and 33 negative ("spam") samples.

a. Using PART algorithm

The following table shows the result obtained using the PART algorithm:

Table. 2 Cross validation results for PART

	Spam	**Ham**	Total
Spam	**19**	**14**	33
Ham	**5**	**162**	167
Total	24	176	200

From this table, we see that 181 messages are correctly classified among 200, and 19 messages are misclassified.

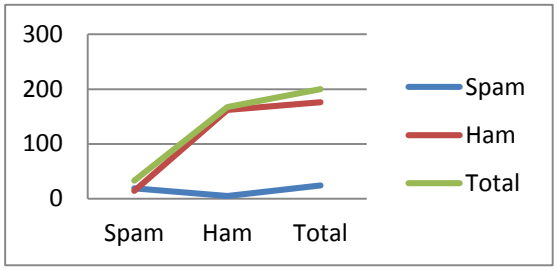


Fig. 4 Number of classified Instances for PART algorithm

or <= 0 AND
to <= 0 AND
2 <= 0: ham (120.0/3.0)
Â£1000 <= 0 AND
FREE <= 0 AND
call <= 0 AND
Reply <= 0 AND
Txt <= 0 AND
Your <= 0 AND
s <= 0 AND
or <= 0 AND
do <= 0 AND
you <= 0: ham (34.0)

i <= 0 AND
me <= 0 AND
Hi <= 0 AND
a > 0: spam (16.0)

i > 0: ham (6.0)

me <= 0 AND
Hi <= 0 AND
and <= 0 AND
I <= 0 AND
the > 0: spam (5.0)

and <= 0 AND
is <= 0: ham (9.0/1.0)

way <= 0: spam (8.0)

: ham (2.0)

Number of Rules : 8

This notation can be read as:

- if (("or" not in message) and ("to" not in message) and ("2" not in message)) then class(message) == ham
- if (("Â£1000" not in message) and ("FREE" not in message) and ("call" not in message)) and ("Reply" not in message)) and ("Txt" not in message)) and ("Your" not in message)) and ("s" not in message)) and ("or" not in message)) and ("do" not in message)) and ("you" not in message)) then class(message) == ham
- if (("i" not in message) and ("me" not in message) and ("Hi" not in message)) and ("a" in message) then class(message) == spam

b. Using Decision Tree algorithm

The following table shows the result obtained using Decision Tree algorithm:

Table. 3 Cross validation results for DT

	Spam	Ham	Total
Spam	**16**	**17**	33
Ham	**5**	**162**	167
Total	21	179	200

From this table, we see that 178 messages are correctly classified among 200, and 22 messages are misclassified.

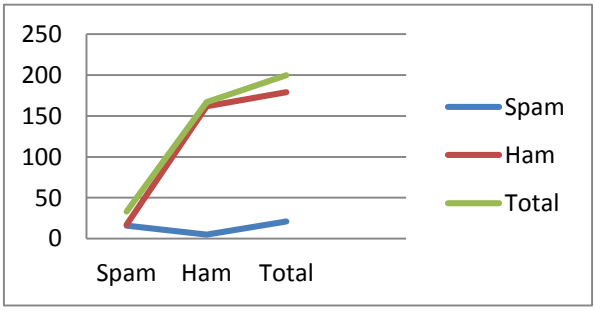


Fig. 5 Number of classified Instances for DT algorithm

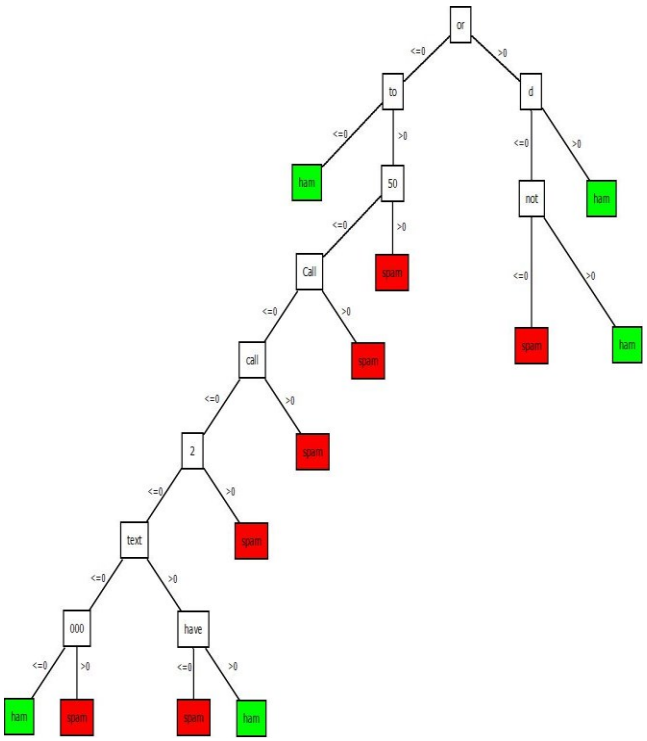


Fig. 3 Decision Tree constructed for the Train Dataset

c. Using NB algorithm

The following table shows the result obtained using Naïve Bayes algorithm:

Table. 4 Cross validation results for Naïve Bayes

	Spam	Ham	Total
Spam	**24**	**9**	33
Ham	**1**	**166**	167
Total	25	175	200

From this table, we see that 190 messages are correctly classified among 200, and 10 messages are misclassified.

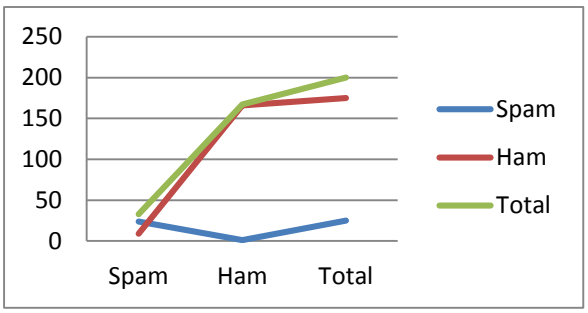


Fig. 6 Number of classified Instances for NB algorithm

d. Using LR algorithm

The following table shows the result obtained using Logistic Regression algorithm:

Table. 5 Cross validation results for Logistic Regression

	Spam	Ham	Total
Spam	**30**	**3**	33
Ham	**4**	**163**	167
Total	44	166	200

From this table, we see that 193 messages are correctly classified among 200, and 7 messages are misclassified.

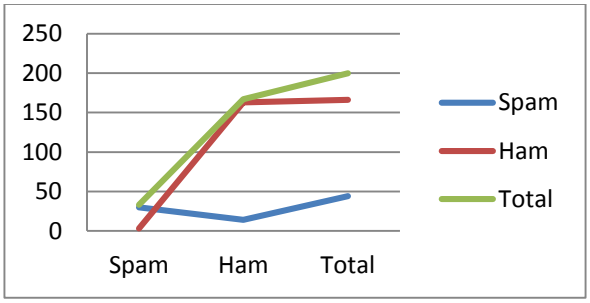


Fig. 7 Number of classified Instances for LR algorithm

e. Using SVM algorithm

The following table shows the result obtained using Support Vector Machine algorithm:

Table. 6 Cross validation results for Support Vector Machine

	Spam	Ham	Total
Spam	**27**	**6**	33
Ham	**2**	**165**	167
Total	29	171	200

From this table, we see that 192 messages are correctly classified among 200, and 8 messages are misclassified.

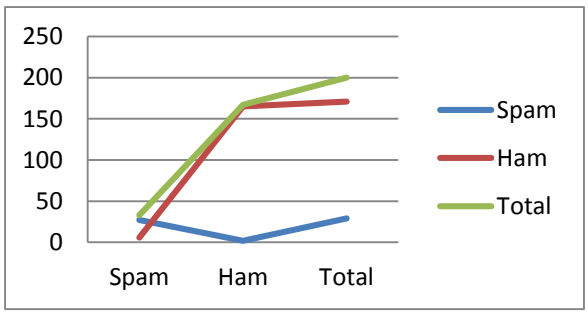


Fig. 8 Number of classified Instances for SVM algorithm

The results in terms of accuracy of the various algorithms used are presented in Table 7.

Table. 7 Number of classified Instances for SMS

	C.C.M	I.C.M	A (%)
PART	181	19	90.5
D.T	178	22	89
N.B	190	10	95
L.R	193	7	96.5
SVM	192	8	96

- C.C.M.: Correctly Classified Messages;
- I.C.M.: Incorrectly Classified Messages;
- A.: Accuracy;

From Table 7 it is evident that Logistic Regression algorithm has highest number of correctly classified instances (96.5%) and Decision Tree algorithm has highest number of incorrectly classified instances (89%).

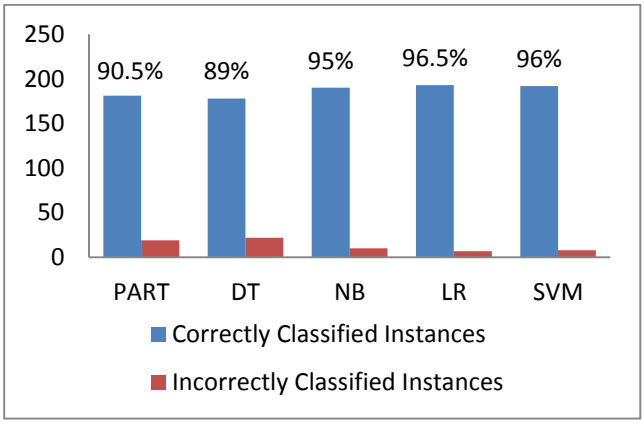


Fig. 9 Number of classified Instances for SMS

From Figure 3 it is evident that Logistic Regression shows the best performance as compare to other studied algorithms. Logistic Regression has highest number of correctly classified instances followed by Support Vector Machine (SVM), Naïve Bayes, PART and Decision Tree.

5. CONCLUSION

Sentiment analysis is essential for anyone who is going to make a decision. It is helpful in different field for calculating, identifying and expressing sentiment. Although the work has yielded interesting results, we plan to make some changes in future work to improve performance and achieve better results. In this paper, we have compared PART, Support Vector Machine, Decision Tree, Naive Bayes, and Logistic Regression algorithms which are very

suitable for generating rules in classification technique. From the experimental results it is concluded that Logistic Regression algorithm seems better than the other four algorithms for SMS Text Messaging Dataset.

6. REFERENCES

[1] Umadevi V. "SENTIMENT ANALYSIS USING WEKA." International Journal of Engineering Trends and Technology (IJETT), Vol.18 (4), Pages 181-183, December 2014.

[2] B. Pang, L. Lee, and S. Vaithyanathan. "THUMBS UP? SENTIMENT CLASSIFICATION USING MACHINE LEARNING TECHNIQUES." In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Pages 79–86, 2002.

[3] Liu, Bing. "SENTIMENT ANALYSIS AND OPINION MINING." Synthesis Lectures on Human Language Technologies Vol.5, no. 1, Pages. 1-167, 2012.

[4] Ahmed, Ishtiaq, Donghai Guan, and Tae Choong Chung. "SMS CLASSIFICATION BASED ON NAÏVE BAYES CLASSIFIER AND APRIORI ALGORITHM FREQUENT ITEMSET." International Journal of Machine Learning & Computing, Vol.4, no. 2, Pages. 183-187, 2014.

[5] Xia, Rui, Chengqing Zong, and Shoushan Li. "ENSEMBLE OF FEATURE SETS AND CLASSIFICATION ALGORITHMS FOR SENTIMENT CLASSIFICATION." Information Sciences Vol.181, Issue 6, Pages 1138–1152, 15 March 2011.

[6] Melville, Prem, Wojciech Gryc, and Richard D. Lawrence. "SENTIMENT ANALYSIS OF BLOGS BY COMBINING LEXICAL KNOWLEDGE WITH TEXT CLASSIFICATION." Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, Pages 1275-1284, 2009.

[7] B.R. Gaines and P. Compton. "INTRODUCTION OF RIPPLE-DOWN RULES APPLIED TO MODELING LARGE DATABASES." Journal of Intelligent Information Systems, Vol.5, Issue 3, Pages 211–228, November 1995.

[8] Aditi Mahajan , Anita Ganpati. "PERFORMANCE EVALUATION OF RULE BASED CLASSIFICATION ALGORITHMS." International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Vol.3, Issue 10, Pages 3546-3550, October 2014

[9] M. Ettaouil, M. Lazaar, K. Elmoutaouakil, K. Haddouch, "A NEW ALGORITHM FOR OPTIMIZATION OF THE KOHONEN NETWORK ARCHITECTURES USING THE CONTINUOUS HOPFIELD NETWORKS", WSEAS TRANSACTIONS on COMPUTERS, Issue 4, Vol.12, Pages 155-163, April 2013.

[10] M. Ettaouil, M. Lazaar, Y. Ghanou, "ARCHITECTURE OPTIMIZATION MODEL FOR THE MULTILAYER PERCEPTRON AND CLUSTERING", Journal of Theoretical and Applied Information Technology. Vol.47, No. 1, Pages 064 – 072, January 2013.

[11] Ali, Shawkat, and Kate A. Smith. "ON LEARNING ALGORITHM SELECTION FOR CLASSIFICATION." Applied Soft Computing, Vol.6, Issue 2, Pages 119-138, January 2006.

[12] Rajaram, Ramasamy, and Appavu Balamurugan. "SUSPICIOUS E-MAIL DETECTION VIA DECISION TREE: A DATA MINING APPROACH." CIT. Journal of computing and information technology, Vol.15, No.2, Pages 161- 169, 2007.

[13] Varsha Sahayak, Vijaya Shete, Apashabi Pathan. "SENTIMENT ANALYSIS ON TWITTER DATA." International Journal of Innovative Research in Advanced Engineering (IJIRAE), Vol.2, Issue 1, Pages 178-183, January 2015.

[14] Nádia F.F. da Silva, Eduardo R. Hruschka, Estevam R. Hruschka Jr. "TWEET SENTIMENT ANALYSIS WITH CLASSIFIER ENSEMBLES." Decision Support Systems, Vol.66, Pages 170–179, October 2014.

[15] Thosten Joachims, "TEXT CATEGORIZATION WITH SUPPORT VECTOR MACHINES: LEARNING WITH MANY RELEVANT FEATURES.", Proceeding of the 10th European Conference on Machine Learning, Pages 137-142, April 21-23, 1998.

[16] Witten, Ian H., and Eibe Frank. "DATA MINING: PRACTICAL MACHINE LEARNING TOOLS AND TECHNIQUES." Morgan Kaufmann, Pages 1-560, June 2005 by Elsevier.