Machine Learning Algorithms for Opinion Mining and Sentiment Classification

Jayashri Khairnar*, Mayura Kinikar**

* Department of Computer Engineering, Pune University, MIT Academy of Engineering, Pune
** Department of Computer Engineering, Pune University, MIT Academy of Engineering, Pune

Abstract- With the evolution of web technology, there is a huge amount of data present in the web for the internet users. These users not only use the available resources in the web, but also give their feedback, thus generating additional useful information. Due to overwhelming amount of user's opinions, views, feedback and suggestions available through the web resources, it's very much essential to explore, analyse and organize their views for better decision making. Opinion Mining or Sentiment Analysis is a Natural Language Processing and Information Extraction task that identifies the user's views or opinions explained in the form of positive, negative or neutral comments and quotes underlying the text. Various supervised or data-driven techniques to Sentiment analysis like Naïve Byes, Maximum Entropy and SVM.For classification use support vector machine (SVM), it performs the sentiment classification task also consider sentiment classification accuracy.

Index Terms- Text mining, support vector machine (SVM), Sentiment Classification, Feature extraction, opinion mining.

I. INTRODUCTION

Text mining offers a way for individuals and corporations to exploit the vast amount of information available on the Internet. In current search engine people to search for other people's opinions from the Internet before purchasing a product or seeing a movie because practically, when we are not familiar with a specific product, we ask our trusted sources to recommend one [6]. Many website provide user rating and commenting services, and these reviews could reflect users' opinions about a product. With the propagation of reviews, ratings, recommendations, and other forms of online expression, online opinion could present essential information for businesses to market their products and manage their reputations. Current search engines can efficiently help users obtain a result set, which is relevant to user's query. However, the semantic orientation of the content, which is very important information in the reviews or opinions, is not provided in the current search engine. For example, Google will return around 7 380 000 hits for the query "Angels and Demons review." If search engines can provide statistical summaries from the semantic orientations, it will be more useful to the user who polls the opinions from the Internet. A scenario for the aforementioned movie query may yield such report as "There are 10 000 hits, of which 80% are thumbs up and 20% are thumbs down." This type of service requires the capability of discovering the positive reviews and negative reviews.Opinion Mining is a process of automatic extraction of knowledge from the opinion of others about some particular topic or problem. This paper will try to focus on the basic definitions of Opinion Mining, analysis of linguistic resources required for Opinion Mining, few machine learning techniques on the basis of their usage and importance for the analysis, evaluation of Sentiment classifications.

Current-day Opinion Mining and Sentiment Analysis is a field of study at the crossroad of Information Retrieval (IR) and Natural Language Processing (NLP) and share some characteristics with other disciplines such as text mining and Information Extraction. Opinion mining is a technique to detect and extract subjective information in text documents. In general, sentiment analysis tries to determine the sentiment of a writer about some aspect or the overall contextual polarity of a document. The sentiment may be his or her judgment, mood or evaluation. A key problem in this area is sentiment classification, where a document is labeled as a positive or negative evaluation of a target object (film, book, product etc.).In recent years, the problem of "opinion mining" has seen increasing attention. Sentiment classification is a recent subdiscipline of text classification which is concerned not with the topic a document is about, but with the opinion it expresses. Sentiment classification also goes under different names, among which opinion mining, sentiment analysis, sentiment extraction, or affective rating.

II. SENTIMENT ANALYSIS

Sentiment analysis of natural language texts is a large and growing field. Sentiment analysis or Opinion Mining is the computational treatment of opinions, sentiments and subjectivity of text. Sentiment analysis is a Natural Language Processing and Information Extraction task that aims to obtain writer's feelings expressed in positive or negative comments, questions and requests, by analyzing a large numbers of documents. Converting a piece of text to a feature vector is the basic step in any data driven approach to Sentiment analysis. Term frequency has always been considered essential in traditional Information Retrieval and Text Classification tasks. But Pang-Lee [1] found that term presence is more important to Sentiment analysis than term frequency. That is, binary-valued feature vectors in which the entries merely indicate whether a term occurs (value 1) or not (value 0).It also reported that unigrams outperform bigrams when classifying movie reviews by sentiment polarity. As a result, the sentiment analysis research from the determination of the semantic orientation of the terms. Determining semantic orientation of words Hatzivassiloglou and McKeown [8] hypothesize that adjectives separated by "and" have the same

polarity, while those separated by "but" have opposite polarity .Starting with small seed lists, this information is used to group adjectives into two clusters such that maximum constraints are satisfied. Sentiment classification is a recent sub discipline of text classification which is concerned not with the topic a document is about, but with the opinion it expresses. Functional to the extraction of opinions from text is the determination of the orientation of "subjective" terms contained in text, i.e. the determination of whether a term that carries opinionated content has a positive or a negative connotation [2]. Esuli and Sebastiani proposed new method for determining the orientation of subjective terms. The method is based on the quantitative analysis of the glosses of such terms, i.e. the definitions that these terms are given in online dictionaries, and on the use of the resulting term representations for semi-supervised term classification. Sentiment classification can be divided into several specific subtasks: determining subjectivity, determining orientation, determining the strength of orientation [2]. Esuli and Sebastiani [4] described SENTIWORDNET, which is a lexical resource in which each WordNet synset is associated with three numerical scores, i.e., Obj(s), Pos(s), and Neg(s), thus describing how objective, positive, and negative the terms contained in the synset.

Traditionally, sentiment classification can be regarded as a binary-classification task [1], [5].Dave,Lawrence,Pennock [5] use structured reviews for testing and training, identifying appropriate features and scoring methods from information retrieval for determining whether reviews are positive or negative. These results perform as well as traditional machine learning method then use the classifier to identify and classify review sentences from the web, where classification is more difficult. Various supervised or data-driven techniques to Sentiment analysis like Naïve Byes, Maximum Entropy and SVM. Pang Lee [1] compared the performance of Naïve Bayes, Maximum Entropy and Support Vector Machines in Sentiment analysis on different features like considering only unigrams, bigrams, combination of both, incorporating parts of speech and position information, taking only adjectives etc. It is observed from the results that:

a. Feature presence is more important than feature frequency.

b. Using Bigrams the accuracy actually falls.

c. Accuracy improves if all the frequently occurring words from all parts of speech are taken, not only Adjectives.

d. Incorporating position information increases accuracy.

e. When the feature space is small, Naïve Bayes performs better than SVM. But SVM's perform better when feature space is increased.

According to their experiment, SVMs tended to do the best, and unigram with presence information turns out to be the most effective feature. In recent years, some researchers have extended sentiment analysis to the ranking problem, where the goal is to assess review polarity on a multipoint scale. Goldberg and Zhu [7] proposed a graph-based semi supervised learning algorithm to address the sentiment-analysis task of rating inference and their experiments showed that considering unlabeled reviews in the learning process can improve rating inference performance.

III. MACHINE LEARNING APPROACHES

The aim of Machine Learning is to develop an algorithm so as to optimize the performance of the system using example data or past experience. The Machine Learning provides a solution to the classification problem that involves two steps:

1) Learning the model from a corpus of training data

2) Classifying the unseen data based on the trained model.

In general, classification tasks are often divided into several sub-tasks:

1) Data preprocessing
2) Feature selection and/or feature reduction
3) Representation
4) Classification
5) Post processing

Feature selection and feature reduction attempt to reduce the dimensionality (i.e. the number of features) for the remaining steps of the task. The classification phase of the process finds the actual mapping between patterns and labels (or targets). Active learning, a kind of machine learning is a promising way for sentiment classification to reduce the annotation cost. The following are some of the Machine Learning approaches commonly used for Sentiment Classification [10].

4.1 **Naive Bayes Classification**

It is an approach to text classification that assigns the class c* = argmaxc P(c | d), to a given document d. A naive Bayes classifier is a simple probabilistic classifier based on Bayes' theorem and is particularly suited when the dimensionality of the inputs are high. Its underlying probability model can be described as an "independent feature model". The Naive Bayes (NB) classifier uses the Bayes' rule Eq. (1),

$$P(c \mid d) = \frac{P(c)P(d \mid c)}{P(d)} \qquad (1)$$

Where, P (d) plays no role in selecting c*. To estimate the term P(d|c), Naive Bayes decomposes it by assuming the fi's are conditionally independent given d's class as in Eq.(2),

$$P_{NB}(c \mid d) = \frac{P(c)\left(\prod_{i=1}^{m} P(f_i \mid c)^{n_i(d)}\right)}{P(d)} \qquad (2)$$

Where, m is the no of features and fi is the feature vector. Consider a training method consisting of a relative-frequency estimation P(c) and P (fi | c). Despite its simplicity and the fact that its conditional independence assumption clearly does not hold in real-world situations, Naive Bayes-based text categorization still tends to perform surprisingly well; indeed, Naive Bayes is optimal for certain problem classes with highly dependent features.

4.2 **Maximum Entropy**

Maximum Entropy (ME) classification is yet another technique, which has proven effective in a number of natural language processing applications. Sometimes, it outperforms Naive Bayes at standard text classification. Its estimate Of P(c | d) takes the exponential form as in Eq. (3) [10],

$$P_{ME}(c \mid d) = \frac{1}{Z(d)} \exp\left(\sum_i \lambda_{i,c} F_{i,c}(d,c)\right) \qquad (3)$$

Where, Z (d) is a normalization function. Fi, c is a feature/class function for feature fi and class c, as in Eq. (4),

$$F_{i,c}(d,c') = \begin{cases} 1 & n_i(d) > 0 \text{ and } c' = c \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

For instance, a particular feature/class function might fire if and only if the bigram "still hate" appears and the document's sentiment is hypothesized to be negative. Importantly, unlike Naive Bayes, Maximum Entropy makes no assumptions about the relationships between features and so might potentially perform better when conditional independence assumptions are not met.

4.3 **Support Vector Machines**

Support vector machines (SVMs) have been shown to be highly effective at traditional text categorization, generally outperforming Naive Bayes. They are large-margin, rather than probabilistic, classifiers, in contrast to Naive Bayes and Maximum Entropy. In the two-category case, the basic idea behind the training procedure is to find a maximum margin hyperplane, represented by vector w, that not only separates the document vectors in one class from those in the other, but for which the separation, or margin, is as large as possible. This corresponds to a constrained optimization problem; letting cj Ɛ {1, −1} (corresponding to positive and negative) be the correct class of document dj, the solution can be written as in Eq. (5) [10],

$$\vec{w} := \sum_j \alpha_j c_j \vec{d}_j, \quad \alpha_j \geq 0 \quad (5)$$

Where, the αj's are obtained by solving a dual optimization problem. That dj such that αj is greater than zero are called support vectors, since they are the only document vectors contributing to w. Classification of test instances consists simply of determining which side of w's hyperplane they fall on.

Support vector machines were introduced in [3] (Vapnik) and basically attempt to find the best possible surface to separate positive and negative training samples. Support Vector Machines (SVMs) are supervised learning methods used for classification. In this project, SVM is used for sentiment classification. First module is sentiment analysis and Support vector machines perform sentiment classification task on review data. The goal of a Support Vector Machine (SVM) classifier is to find a linear hyperplane (decision boundary) that separates the data in such a way that the margin is maximized. Look at a two class separation problem in two dimensions like the one illustrated in figure 1, observe that there are many possible boundary lines to separate the two classes. Each boundary has an associated margin. The rationale behind SVM's is that if we choose the one that maximizes the margin we are less likely to misclassify unknown items in the future.

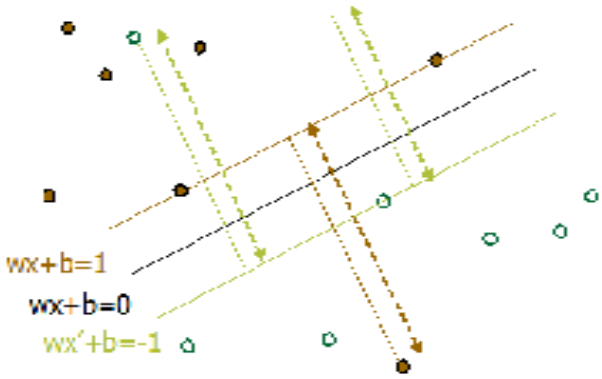

Figure 1: Different boundary decisions are possible to separate two classes in two dimensions. Each boundary has an associated margin.

What SVM is used for?

SVM is primarily used for categorization. Some examples of SVM usage include bioinformatics, signature/hand writing recognition, image and text classification, pattern recognition, and e-mail spam categorization. Many research documents such as the ones mentioned above have shown that SVM can classify reasonably well. In this project, SVM is used for text classification. Text classification is a method used to put text into meaningful groups. Besides SVM, there are many other methods for text classification such as Bayes and k-Nearest Neighbor. Based on many research papers (Joachims, T., 1998), SVM outperforms many, if not all, popular methods for text classification. The studies also show that SVM is effective, accurate, and can work well with small amount of training data [12].

How SVM Works

The idea for SVM is to find a boundary (known as a hyperplane) or boundaries that separate clusters of data. SVM does this by taking a set of points and separating those points using mathematical formulas. The following figure illustrates the data flow of SVM.

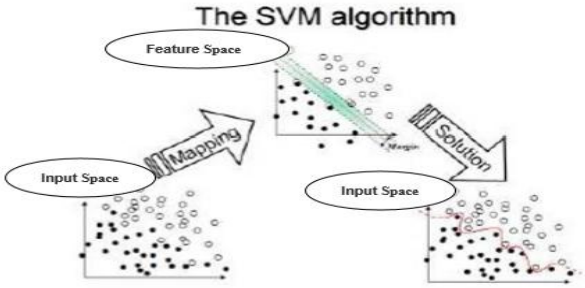

Figure 2: SVM Process Flow

In Figure 2, data are input in an input space that cannot be separated with a linear hyperplane. To separate the data linearly, points are map to a feature space using a kernel method. Once the data in the feature space are separate, the linear hyperplane gets map back to the input space and it is shown as a curvy non-

linear hyperplane. This process is what makes SVM amazing. The SVM's algorithm first starts learning from data that has already been classified, which is represented in numerical labels (e.g. 1, 2, 3, etc.) with each number representing a category. SVM then groups the data with the same label in each convex hull. From there, it determines where the hyperplane is by calculating the closest points between the convex hulls (Bennett, K. P., & Campbell, C., 2000). Once SVM determines the points that are closest to each other, it calculates the hyperplane, which is a plane that separates the labels.

Simple SVM Example

Let us use a few simple points to illustrate the concept of SVM. The following example is similar to Dr. Guestrin's lecture (Guestrin, C., 2006). Given the following points with corresponding classes (labels) in Figure 3, find a hyperplane that separated the points [12].

Table 1: Simple Data in 1-Dimension

Class	X_1
+1	0
-1	1
-1	2
+1	3

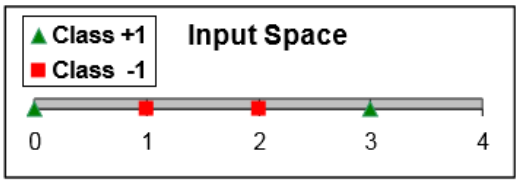

Figure 3: Simple Data in an Input Space

As Figure 3 shows, these points lay on a 1-dimensional plane and cannot be separated by a linear hyperplane. The first step is to find a kernel that maps the points into the feature space, then within the feature space, find a hyperplane that separates the points. A simple kernel that would do the trick is $\Phi(X1) = (X1, X12)$. This kernel is actually a polynomial type. As the reader sees, this kernel will map the points to a 2-dimensional feature space by multiplying the points to the power of 2. From calculating the kernels, we get (0, 0, +1), (1, 1, -1), (2, 4, -1), (3, 9, +1) [12].

Table 2: Simple Data in 2-Dimension

Class	X_1	X_1^2
+1	0	0
-1	1	1
-1	2	4
+1	3	9

Figure 4: Simple Data in a Feature

The next step is finding a hyperplane

$\langle w \bullet x \rangle + b = +1$ (positive labels) (1)

$\langle w \bullet x \rangle + b = -1$ (negative labels) (2)

$\langle w \bullet x \rangle + b = 0$ (hyperplane) (3)

From these equations, find the unknowns, w and b. Expanding the equations for the SVM problem will get:

w1x1 + w2x2 + b = +1

w1x1 + w2x2 + b = -1

w1x1 + w2x2 + b = 0

Solve w and b for the positive labels using equation, w1x1 + w2x2 + b = +1.

w1x1 + w2x2 + b = +1

→ w10 + w20 + b = +1

→ w13 + w29 + b = +1

Solve w and b for the negative labels using equation, w1x1 + w2x2 + b = -1.

w1x1 + w2x2 + b = -1

→ w11 + w21 + b = -1

→ w12 + w24 + b = -1

By using linear algebra, we find that the solution is w1 = -3, w2 = 1, b = 1, which satisfies the above equations. Many times, there is more than one solution or there may be no solution, but SVM can find the optimal solution that returns a hyperplane with the largest margin. With the solutions: w1 = -3, w2 = 1, b = 1, positive plane, negative plane, and hyperplane can be calculated.

Table 3: Calculation Results of Positive, Negative, and Hyperplane

Positive Plane:	Negative Plane:	Hyperplane:
$\langle w \bullet x \rangle + b = +1$	$\langle w \bullet x \rangle + b = -1$	$\langle w \bullet x \rangle + b = 0$
$w_1x_1 + w_2x_2 + b = +1$	$w_1x_1 + w_2x_2 + b = -1$	$w_1x_1 + w_2x_2 + b = 0$
→ $-3x_1 + 1x_2 + 1 = +1$	→ $-3x_1 + 1x_2 + 1 = -1$	→ $-3x_1 + 1x_2 + 1 = 0$
→ $x_2 = 3x_1$	→ $x_2 = -2 + 3x_1$	→ $x_2 = -1 + 3x_1$

X_1	X_2
0	0
1	3
2	6
3	9

X_1	X_2
0	-2
1	1
2	4
3	7

X_1	X_2
0	-1
1	2
2	5
3	8

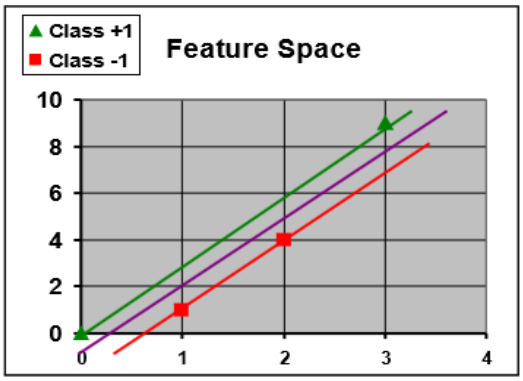

Figure 5: Simple Data in a Feature Space Separated by a Hyperplane

Thus, we have the model that contains the solution for w and b and with margin $2/\sqrt{(w \bullet w)}$. The margin is calculated as follow.

$2/\sqrt{(w \bullet w)}$ (4)

$2/\sqrt{(-32 + 12)}$ → margin = 0.632456

In SVM, this model is used to classify new data. With the solutions, new data can be classified into category. For example, if the result is less than or equal -1, the new data belongs to the -1 class and if the result is greater than or equal to +1, the new data belongs to the +1 class.

LIBSVM is a well-known library for SVM that is developed by Chih-Chung Chang and Chih-Jen Lin. LIBSVM is a library for Support Vector Machines (SVMs).LIBSVM is an integrated software for support vector classification, (C-SVC, nu-SVC), regression (epsilon-SVR, nu-SVR) and distribution estimation (one-class SVM) [9]. It supports multi-class classification. LIBSVM involves two steps: first, training a data set to obtain a model and second, using the model to predict information of a testing data set. SVM procedure includes Transform data to the format of an SVM package, Conduct simple scaling on the data, Select model here use linear formula, Use cross-validation to find the best parameter, Use the best parameter to train the whole training set and Test.

IV. EVALUATION OF SENTIMENT CLASSIFICATION

In general, the performance of sentiment classification is evaluated by using four indexes. They are Accuracy, Precision, Recall and F1-score [11]. The common way for computing these indexes is based on the confusion matrix as shown below:

Table 4: Confusion Matrix

#	Predicted positives	Predicted negatives
Actual positive instances	Number of True Positive instances (TP)	Number of False Negative instances (FN)
Actual negative instances	Number of False Positive instances (FP)	Number of True Negative instances (TN)

These indexes can be defined by the following equations:

- $\text{Accuracy} = \dfrac{TN + TP}{TN + TP + FP + FN}$
- $\text{Precision} = \dfrac{TP}{TP + FP}$
- $\text{Recall} = \dfrac{TP}{TP + FN}$
- $\text{F1} = \dfrac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$

Accuracy is the portion of all true predicted instances against all predicted instances. An accuracy of 100% means that the predicted instances are exactly the same as the actual instances. Precision is the portion of true positive predicted instances against all positive predicted instances. Recall is the portion of true positive predicted instances against all actual positive instances. F1 is a harmonic average of precision and recall.

V. CONCLUSION

Some of the machine learning techniques like Naïve Bayes, Maximum Entropy and Support Vector Machines has been discussed. Many of the applications of Opinion Mining are based on bag-of-words, which do not capture context which is essential for Sentiment Analysis. The recent developments in Sentiment Analysis and its related sub- tasks are also presented. The state of the art of existing approaches has been described with the focus on the Sentiment Classification using various Machine learning techniques. This paper introduced and surveyed the field of sentiment analysis and opinion mining. It has been a very active research area in recent years. In fact, it has spread from computer science to management science. Finally, this paper concludes saying that all the sentiment analysis tasks are very challenging. The concept of SVM is explained through a small set of data in a 2-dimenional feature space. With the use of kernel methods, SVM can classify data in high dimensional space. SVM is an excellent method for data classification. Finally, the future challenges and directions so as to further enhance the research in the field of Opinion Mining and Sentiment Classification are discussed.

REFERENCES

[1] B. Pang, L. Lee, and S. Vaithyanathan,"Thumbs up?:Sentiment classification using machine learning techniques,"inProc.ACL-02Conf.Empirical Methods Natural Lang. Process., 2002, pp. 79–86.

[2] A. Esuli and F. Sebastiani, "Determining the semantic orientation of terms through gloss classification," in Proc. 14th ACM Int. Conf. Inf. Knowl.Manage., 2005, pp. 617–624.

[3] V. N. Vapnik, The Nature of Statistical Learning Theory.New York:Springer-Verlag, 1995.

[4] A. Esuli and F. Sebastiani, "SENTIWORDNET: A publicly available lexical resource for opinion mining," in Proc. 5th Conf. Lang. Res. Eval., 2006, pp. 417–422.

[5] K. Dave, S. Lawrence, and D. M. Pennock, "Mining the peanut gallery:opinion extraction and semantic classification of product reviews," in Proc. 12th Int. Conf. World Wide Web, New York: ACM, 2003, pp. 519–528.

[6] Chien-Liang Liu, Wen-Hoar Hsaio, Chia-Hoang Lee, Gen-Chi Lu, and Emery Jou, "Movie Rating and Review Summarization in Mobile Environment",IEEE VOL. 42, NO. 3, MAY 2012

[7] A. B. Goldberg and X. Zhu, "Seeing stars when there aren't many stars:Graph-based semi-supervised learning for sentiment categorization," in Proc. TextGraphs: First Workshop Graph Based Methods Nat. Lang. Process, Morristown, NJ: Assoc. Comput. Linguist. 2006, pp. 45–52.

[8] V. Hatzivassiloglou and K. R. McKeown, "Predicting the semantic orientation of adjectives," in Proc. 8th Conf. Eur. Chap. Assoc. Comput.Linguist., Morristown, NJ: Assoc. Comput. Linguist, 1997, pp. 174–181.

[9] (2001). LIBSVM: A library for support vector machines [Online]. Available: http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[10] S. ChandraKala and C. Sindhu ISSN: 2229-6956(ONLINE) ICTACT JOURNAL ON SOFT COMPUTING, OCTOBER 2012, VOLUME: 03, ISSUE: 01 "OPINION MINING AND SENTIMENT CLASSIFICATION: A SURVEY". \

[11] International Journal of Ad hoc, Sensor & Ubiquitous Computing (IJASUC) Vol.4, No.1, February 2013, "Opinion Mining and Sentiment Analysis –An Assessment of Peoples' Belief: A Survey"S Padmaja and Prof. S Sameen Fatima.

[12] "Clustering High Dimensional Data Using SVM"Tam P. Ngo,December 2006.

AUTHORS

First Author – Jayashri Khairnar received her Bachelor's degree in Information Technology .Now; she is pursuing her M.E degree in Computer Engineering from MIT Academy of Engineering, Pune University, Pune, India. Her research areas are Data mining and Text mining. Email- jaynit15@gmail.com

Second Author – Prof. Mayura Kinikar, B.E., M.E. Computer was educated at Doctor Babasaheb Ambedkar Marathwada University. Now, she is pursuing her PhD. She has worked in various capacities in academic institutions. Now she is Assistant Prof in MIT Academy of Engineering, Alandi, Pune. Her areas of interest include Data mining, text mining, web mining and warehousing.
Email-mukinikar@comp.maepune.ac.in.