Comparison of Naive Basian and K-NN Classifier

Deepak Kanojia
Me (Cse)
Tieit, Bhopal

Mahak Motwani
Assistant Professor (Cse Department)
Tieit, Bhopal

ABSTRACT

In this paper comparison is done between k-nearest neighbor and naïve basin classifier based on the subset of features. Sequential feature selection method is used to establish the subsets. Four categories of subsets are used like life and medical transcripts, arts and humanities transcripts, social science transcripts, physical science transcripts to show the experimental results to classify data and to show that K-NN classifier gets competition with naïve basian classifier. The classification performance K-NN classifier is far better then naïve basian classifier when learning parameters and number of samples are small. But as the number of samples increases the naïve basian classifier performance is better K-NN classifier. On the other hand naïve basian classifier is much better then K-NN classifier when computational demand and memory requirements are considered. This paper demonstrates the strength of naïve basian classifier for classification and summarizes the some of the most important developments in naïve basian classification and K- nearest neighbor classification research. Specifically, the issues of posterior probability estimation, the link between Naïve basian and K-NN classifiers, learning and generalization tradeoff in classification, the feature variable selection, as well as the effect of misclassification costs are examined. The purpose is to provide a synthesis of the published research in this area and stimulate further research interests and efforts in the identified topics.

General Terms

Semi supervised text classification, naïve basian, K-NN, text mining.

Keywords

Naïve Bayesian classifier, K-NN classifier classification, ensemble methods, feature variable selection, learning and generalization, misclassification costs, k-means clustering.

1. INTRODUCTION

It is always a big question to use two techniques, information retrieval and text mining in a efficient way to utilize huge amount of data. When document clustering is used, text mining works in an efficient way. Document clustering involves the process of organizing the text documents, summarizing the text documents in an efficient way so that meaningful clusters can be created to organize the massive amount of documents. Once the query is generated on behalf of users to get organized result by searching with the help of search engine and will show the number of documents in document clustering and it will improve the accuracy and will show the improved output for information retrieval system [1]. This is one of the good method with the help of which gives the nearest-neighbor of a document [2].the problems which are of different types can be given as: if a group of documents are given and the number of clusters is to be find that can partition the documents in a manner that is predefined and automatically. The document clustering is done in such a manner that the documents those are similar can be put in same cluster and those are different can be put on different clusters or the documents those belongs to same topic are assigned to same clusters and those belongs to other topics can be put on different clusters in most of the document clustering algorithms vector space model [1] is used Where a document is treated as a bag of words. If dimensionality is high it will be treated as a good property for feature space. On behalf of this representation a big question is marked on the performance of clustering algorithm. When high dimensional feature space is used can not get the efficiency because of inherent sparseness of the data [3]. One of the other problem that is that in document clustering all the features can not be used because some of the features are redundant and irrelevant or some features can affect the clustering results specially when condition is that relevant features are few than that of irrelevant features in such a situation when original features of subsets are used it gives the better performance [4]. The high dimensionality of feature space is reduced when feature selection is used while on the other hand the better understanding of data can be get and through which the performance of clustering result can be improved. Features those are used for clustering should contain reliable and sufficient information of original data set. While document clustering is considered the above properties are formulated in to the problem to extract the information words with in a set of documents. While considering the supervised learning the feature selection is widely used for text classification. The redundant and irrelevant terms are removed by feature selection process from the data corpus through which the accuracy and efficiency can be improved in text classification process [5]. Supervised learning and unsupervised learning are two classification processes those are used for feature selection. It will depend on which type of class label information is needed for each document. Term strength and document frequency can be applied easily to clustering if one uses in unsupervised feature selection methods [6]. But we can see in [4] that using information gain in unsupervised feature selection methods gives better accuracy [7]. The supervised feature selection methods can not applied generally to document clustering because required class label information is not available.

1.1 Naïve-Basian Classifier

Class label those are conditionally independent to each other based on attributes is the main assumption of naïve basian (NB) decision rule[8] although independence assumption is considered between attributes apart from that the naïve basian performs well amazingly even though this condition is not interested in consideration for most of the data sets as the literature [9] is considered. The independence property of data set shows that although any relation exists in between the data sets, it will be totally ignored while considering the naïve basian classifier. If the class label of any node is Ω and it has two attribute values as Xi and Xj (where Xi≠Xj) and are conditionally independent. Then x_i will be conditionally independent of x_j for given class ω. When the condition of $P(x_i\backslash\omega, x_j)= P(x_j, \omega)$ is met for all values of $x_i \in Xi$ and $x_j \in Xj$ where $\omega \in \Omega$ and for condition $P(x_j, \omega)>0$ is met. Naïve basian classifier structure is shown in fig: 1 in the form of basian network. The irrelevant features are removed, by using search algorithm then involve the feature selection method and this form of method is known as selective naïve basian classifier (SNB). According to given figure of naïve basian classifier structure represents that every attribute is not depend on any other remaining attribute whose class label is given as ω of the class variable. The root node of tree whose value is Ω works as a parent node for each attribute values Xi can be represented as $\prod_{xi} =\{\Omega\}$ for all values $1 \leq i \leq n$. for the same structure the joint probability distribution of $P(X_{1,} X_{2\ldots}Xn, \Omega)$ for the same network can be determined as $P(X_{1,} X_{2\ldots}Xn, \Omega)=\prod_{i=1}^{n+1} P(U_i\backslash\prod_{ui})=P(\Omega)\prod_{i=1}^{n} P(X_i\backslash\Omega)$,and according to the definition of conditional probability for the classes that belongs to Ω can have the value of attribute as $P(\Omega\backslash X_{1,} X_{2\ldots}Xn)= \alpha P(\Omega) \prod_{i=1}^{n} P(X_i\backslash\Omega)$ where α is normalization constant.

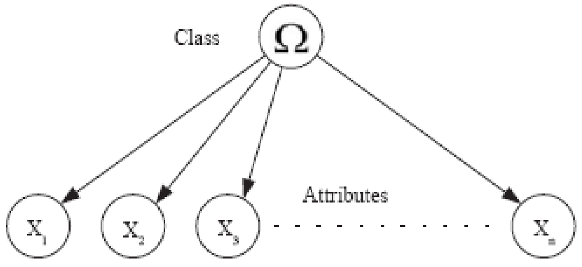


Fig: 1 structure of naïve basian network

1.2 K-NN Classifier

In 1968, Cover and Hart proposed an algorithm the K-Nearest Neighbor [10], which was maturely developed after some time. K-Nearest Neighbor can be calculated by calculating Euclidian distance, although other measures are available, through Euclidian distance we have fine mixing of ease and efficiency. The example is classified by determining the majority of votes of the labels for K-Near neighbor [11]. In other words this method is very easy to implement for instance if an example x has k nearest examples where feature space and majority of them are having the same label y, then x belongs to y. the K-NN method is mostly depend upon utmost theorem while considering theory . When the decision course is considered consider small number of nearest neighbor. Hence when this method is used, example imbalanced problem can be solved. While limited number of nearest neighbor are considered by K-NN, not a decision boundary. Hence good to say that K-NN is suitable to classify the case of example set of boundary intercross and in that case example overlapped. The Euclidian distance can be calculated as follows [12]. If two vectors x_i and x_j are given where $x_i = (x_i^1 , x_i^2 , x_i^3 \ldots\ldots\ldots\ldots x_i^n)$ And $x_j=(x_j^1, x_j^2, x_j^3\ldots\ldots\ldots\ldots x_j^n)$ The difference between x_i and x_j is $D (x_{i,} x_{j)} =\sqrt{\sum_{k=1}^{n}((x_i^k - x_j^k)^2}$ [13]. In this experiment, this formula is used to estimate the nearest neighbor of an example. The K-NN algorithm is very effective and simple to implement. While one of the main drawback of K-NN is its inefficiency for large scale and high dimensional data sets. The main reason of its drawback is its "lazy" learning algorithm natures and it is because it does not have a true learning phase and that results a high computational cost at the classification time. Yang and Liu [14] set k as 30-45 since they found stable effectiveness in those range. In the same way Joachims [15] tried over different k∈ {15, 30, 45, 60}. When the above two attempts are considered, k values are explored, where k∈ {15,30,45} for the K-NN classifier and have the best performance for the value of 'k' that results on the test samples as shown in figure. The K-NN classifier (also known as instance based classifier) perform on the premises in such a way that classification of unknown instances can be done by relating the unknown to the known based on some distance/similarity function. The main objective is that two instances far apart in the instance space those are defined by the appropriate distance function are less similar than two nearly situated instances to belong to the same class [16].

2. COMPARATIVE STUDY OF NAÏVE BASIAN AND K-NN CLASSIFIER

A binary transformation is less suitable for transformation rather than any other transformation because if a word is present or absent from the documents then it reduces its values from frequency to binary variables on the basis of previous study it is found that binary transformation is worst in performance while on the other hand inverse document frequency is used because it decrease the importance of any word among the collection of document. In this experiment this is to show that when these characteristics are used they improved the performance slightly. Any other characteristics for the performance are not used.

2.1 Comparative performance behavior

As shown in the figure when nearest neighbor and naïve basin classifier are used shows a good status when data sets training is started with small number of documents and when the number of documents increases the difference starts showing the performance of these two classifiers differs. For the larger training sets the performance of naïve basian is much better than KNN classifier but it does not seems that if features are increasing then the performance drops.

2.2 Comparative processing time behavior

When processing time is considered it is shown that the processing time is totally depend upon the size of test set as the size increases the processing time increases and remain same for these two classifiers and if different number of documents (and of different test size) are used then we can observed the processing time differences. As the number of features changes the training time for both classifiers is to train data is required much and as the number of features increases the time required to train data set is less.

3. RAPID MINER

RapidMiner early known as Yet Another Learning Process(YALE) and is a tool for text mining, data mining, machine learning and predictive analysis and can be used for training, education, research work, industrial application, application development and rapid prototyping. RapidMiner is given a rank second for real projects in 2009 and ranked first in 2010 for data mining analytical tools according to a poll done by a data mining news paper, KDnuggets. It is hosted by Source Forge since 2004 and is an open source license, distributed under the AGPL. Ralf Klinkenberg, Ingo Mierswa and Simon Fischer started the RapidMiner Project in 2001 at the Artificial Intelligence Unit of the Dormund University Technology. The procedures that can have through RapidMiner are machine learning procedures that include data transformation and data loading (Extract, Transform, Load a.k.a. ETL) and data visualization and data modeling, data evaluation and deployment. The rapid miner tool is totally written in Java Programming Language. The learning procedure of rapid miner for statistical modeling schemes are done through R-project and attribute evaluator's schemes from the Weka machine learning environment. The analytical steps (similar to R) can be defined through RapidMiner and is used for analyzing the data those are generated by high-throughput machines those are used for the purpose like genotyping, proteomics and mass spectrometry. RapidMiner can also be used for the following process like data stream engineering, feature engineering, multimedia mining, text mining, and development of ensembles methods and distributed data mining. Additional plugin's can be added to enhance the functionality of RapidMiner. RapidMiner's graphical user interface environment is provided by RapidMiner that gives the feature to analytical pipeline design environment or the "Operator Tree". Through GUI and XML (eXtensible Markup Language) can be generated to the file that defines the analytical processing where the user wishes to apply to the data or in other words to other programs can call the engine or can be used as an API. The function can be individually called from the command line. RapidMiner is an open source software and is free from charge as a community edition released under the GNU AGPL. It also offers an Enterprise edition under a commercial license for integration into closed source projects different services in the field of text mining, data mining, predictive analysis for software solutions and services are provided by Rapid-I. Automatic intelligent analysis on large scale base is the main concentration of the company, i.e. for large amount of unstructured data like texts, and structured data like database system. For data mining and business intelligence application's the open source data mining system RapidMiner-I provides an environment to use it in efficient way. From existing data sets to extract and discover the unused business intelligence and to provide better informed decisions and to provide the process optimization is the specialty of Rapid-I. it works as a data mining engine and is available as a stand alone application for data analysis and can be integrated into its own products. Till now, more than 30 countries and thousands of RapidMiner applications are giving a comprehensive edge to its users. The well known companies like Cisco, HP, IBM, Philips , Miele, Nokia, Honda, Ford, Merrill Lynch, BNP Paribas, Bank of America, mobikom austria, Akzo nobel, Aureus Pharma, Pharma DM, Revere, Celera, Cyprotex, LexisNexis, Mitre are amons the users and many medium-sized businesses those are benefitting from the Open-Source business model of Rapid-I or unquestionably one of the open source world leading software is RapidMiner for data mining system and is a standalone application for data mining and data analysis.

3.1 Overview

Few processes like data analysis, data integration, analytical ETL and reporting is one single suite powerful but intuitive graphical user interface to provide the better analysis and design of the processes and to provide repositories for the processes and to handle the metadata and data and to provide the solution for the problem of meta data transformation. Forget trial and error and it inspect results already during design time. It gives only the solution which supports on-the-fly recognition and provides the interface to quick fix complete and flexible of data sets and also gives the feature to data loading, data transformation methods.

3.2 Features

It is one of the open source data analysis system for data mining application and can be run on every operating system and on every platform and is freely available. Most intuitive process design multi layered data view concepts gives the feature to ensure efficient data handling on GUI mode, server mode (command line) and provide the process to access via Java API simple Extension mechanism and powerful high-dimensional plotting facilities. It provides lots of features and most comprehensive solutions are available like it provides more then 500 operators for data transformation and data integration, data evaluation, data visualization and data mining. It also provides automatic Meta optimization schemes for data. It gives the definition of reusable building blocks and standardized XML interchange format for processes. The other features includes graphical process design for standard task and to give scripting language for arbitrary operations, machine learning library WEKA fully integrated Access to data sources like Excel, Access, Oracle, IBM DB2, Sybase, MySQL, Ingres, Text file and more. It also gives the features to most comprehensive data mining solutions with respect to data integration, data transformation and data modeling methods. It is also the winner of several users and jury awards.

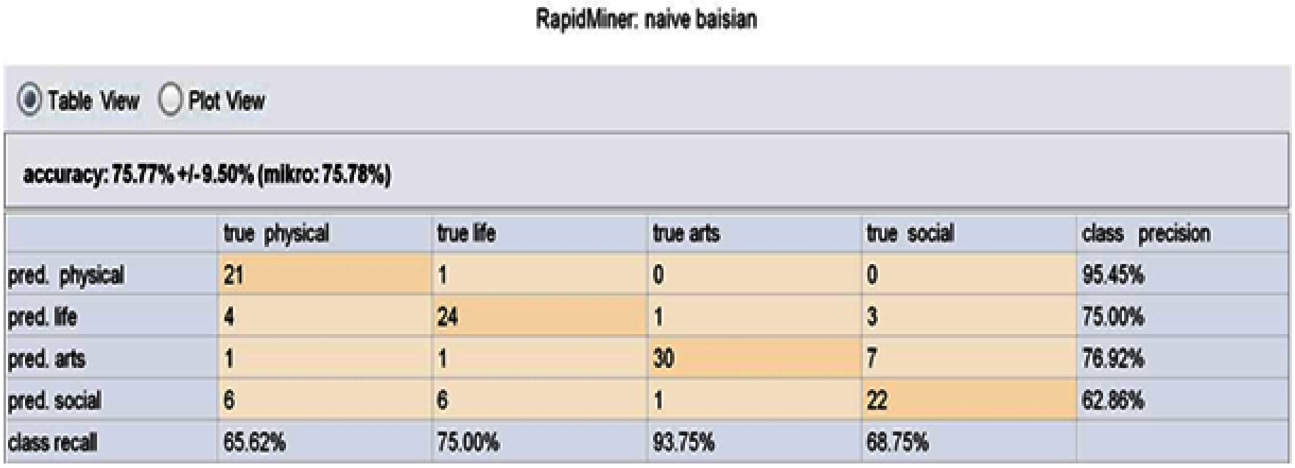
4. PROPOSED WORK

The proposed work is to classify the data of four categories Life and medical science transcripts, arts and humanities transcripts, social science transcripts, physical science transcripts. While working chose 80% of the data to train the data using naïve basian classification technique and K nearest neighbor classification techniques. Once we trained the data we use 20% of the rest of data to compute the categories of the data whether which belongs to which categories. This work is computed among these two categories which classification technique is good and gives more accuracy following with these two categories.

Fig: 2 General tool view of RapidMiner

5. EXPERIMENTAL RESULT

5.1 Naïve basian for known data set: Here the accuracy table shows the 80% of the data when classified using naïve basian classification give the accuracy of which level.



RapidMiner: naive baisian

accuracy: 75.77% +/- 9.50% (mikro: 75.78%)

	true physical	true life	true arts	true social	class precision
pred. physical	21	1	0	0	95.45%
pred. life	4	24	1	3	75.00%
pred. arts	1	1	30	7	76.92%
pred. social	6	6	1	22	62.86%
class recall	65.62%	75.00%	93.75%	68.75%	

Fig 3: Table view of naïve basian for known data set

5.4 Plot view for unknown data set: once the data is classified by training classifier generated plot view shows the file of unknown data set those belong to four categories data set

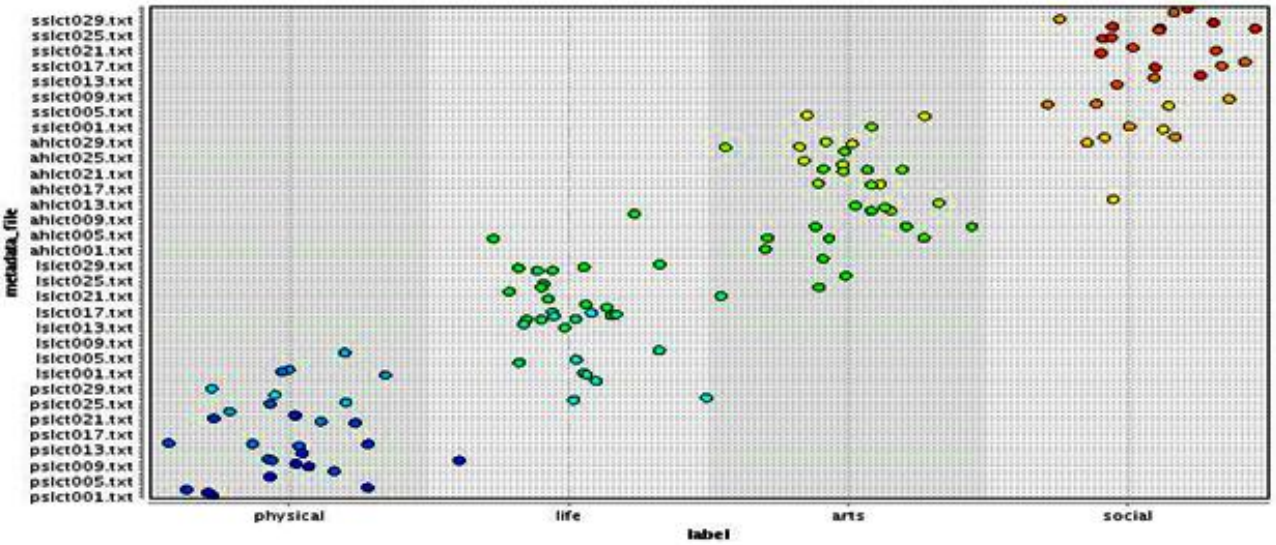


Fig4: Plot view of naïve basian for unknown data set

5.5 Tree structure for known data set: Here tree structure for known data set t shows corresponding cluster nodes to which known dataset documents belongs.

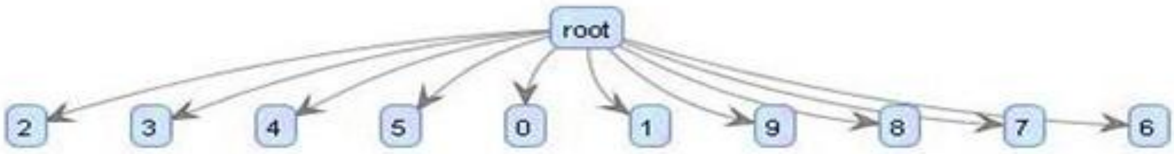


Fig 5: Tree structure of naïve basian for known data set

5.6 Cluster view of unknown data set: once the unknown data is classified their cluster view is given by the given figure below.

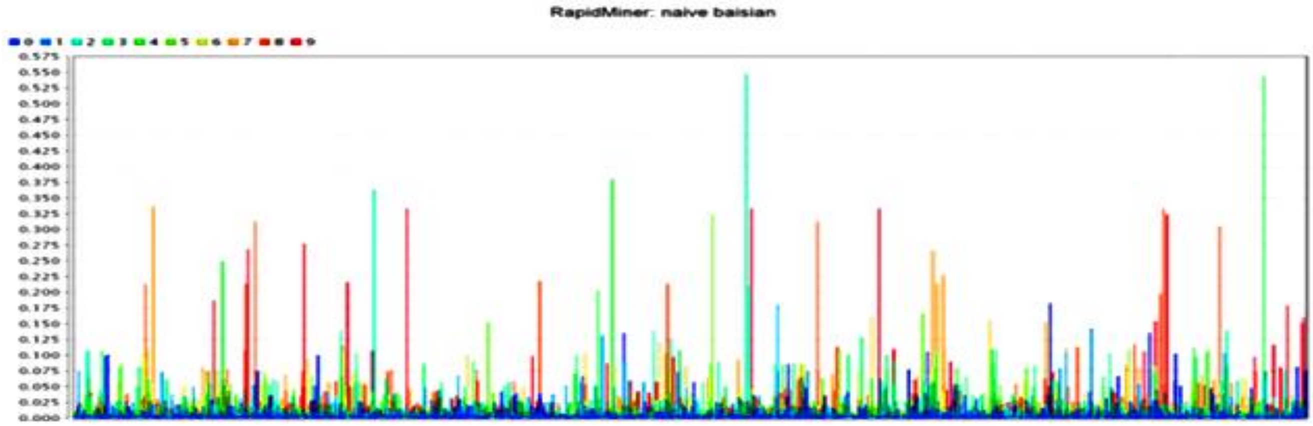
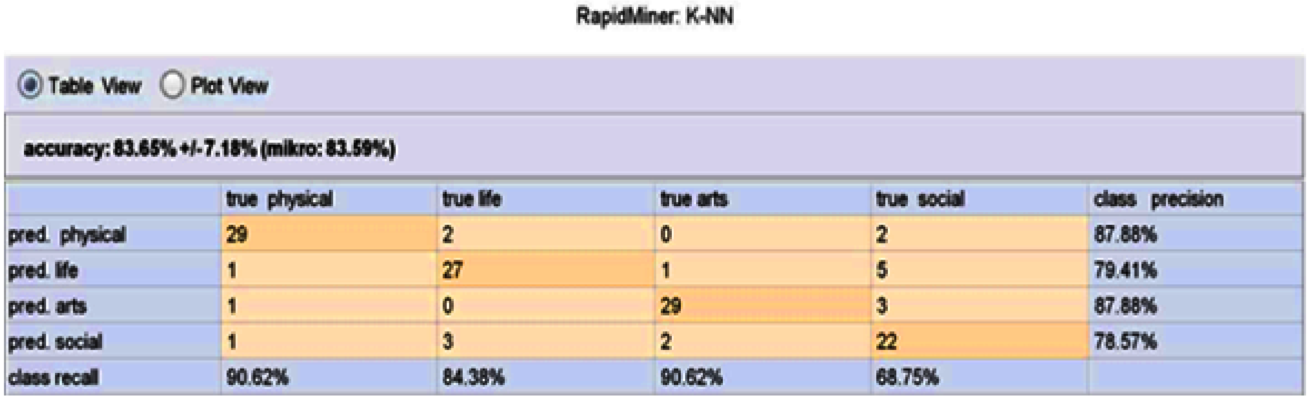


Fig 6: Cluster view of naïve basian for unknown data set

5.7 K-NN for known data set: Here accuracy table shows the 80% of the data when classified using K-nearest neighbor classification give the accuracy of which level.



accuracy: 83.65% +/-7.18% (mikro: 83.59%)

	true physical	true life	true arts	true social	class precision
pred. physical	29	2	0	2	87.88%
pred. life	1	27	1	5	79.41%
pred. arts	1	0	29	3	87.88%
pred. social	1	3	2	22	78.57%
class recall	90.62%	84.38%	90.62%	68.75%	

Fig 7: table view of naïve basian for known data set

5.10 Tree structure for known data set: Here tree structure for known data set shows corresponding cluster nodes to which known dataset documents belongs.(figure is same as figure-5.5)

5.11 Plot view for unknown data set: once the data is classified by training classifier generated plot view shows the file of unknown data set those belong to four categories data set.

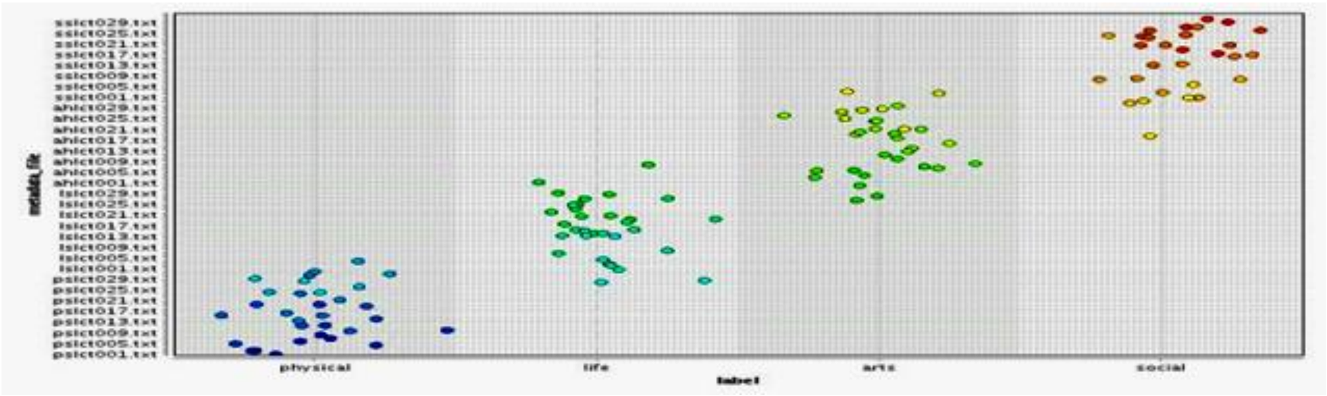


Fig 8: Plot view of naïve basian for unknown data set

5.12 Cluster view of unknown data set: once the unknown data is classified their cluster view is given by the given figure below.

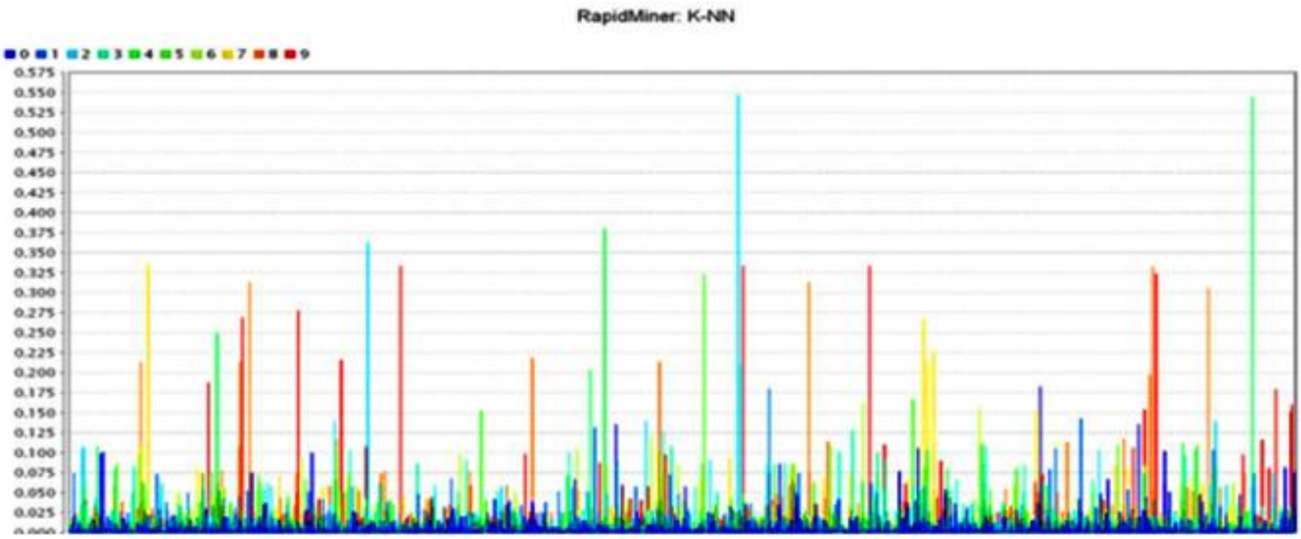


Fig 9: Cluster view of naïve basian for unknown data set

6. CONCLUSION

As in the study, on the basis of literature result it is found that naïve basian classifier works as best document classifier when implementation is done on the basis of different features selection techniques and classification available in RapidMiner When these two classification methods are applied on same data sets to find the optimal result shows that K-NN classification method gives more accuracy (approx 83.65%) as compare to naïve basian classification method that gives the accuracy result (approx 75.77%) . the classification can be further be improved by incorporating various other attributes and increasing the number of cases for training and testing. The efficiency of result can be further increased by using better feature selection methods like CHI Square, Relevance Factor, Information Gain and other weighted features.

7. ACKNOWLEDGMENTS

Specially thankful to my guide Ms. Mahak Motwani, Assistant Professor at Truba Institute of Engineering & Information Technology who helped me and guided me to complete this paper. Hearty thanks to worthy institute TIEIT, Bhopal and all those that we have inadvertently forgotten for their immense support and encouragement.

8. REFERENCES

[1] C. J. van Rijsbergen, *Information Retrieval*, 2nd edition, Butterworth, London, 1979.

[2] C. Buckley and A. F. Lewit, “Optimizations of Inverted Vector Searches,” Proc. of Annual ACM SIGIR Conf. on Research and Development in Information Retrieval, pp. 97–110, 1985.

[3] C. C. Aggrawal and P. S. Yu, “Finding Generalized Projected Clusters in High Dimensional Spaces,” Proc. of ACM SIGMOD Int’l Conf. on Management of Data, pp. 70–81, 2000.

[4] T. Liu, S. Liu, Z. Chen, and W. Ma, “An Evaluation on Feature Selection for Text Clustering,” Proc. of Int’l Conf. on Machine Learning, 2003.

[5] Y. Yang and J. O. Pedersen, “A Comparative Study on Feature Selection in Text Categorization,” Proc. of Int’l Conf. on Machine Learning, pp.412–420, 1997.

[6] Y. Yang, “Noise Reduction in a Statistical Approach to Text Categorization,” Proc. of Annual ACM SIGIR Conf. on Research and Development in Information Retrieval, pp. 256–263, 1995.

[7] J. R. Quinlan, “Induction of Decision Trees,” Machine Learning, vol. 1, pp. 81–106, 1986.

[8] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, Wiley, New York, 2000.

[9] N. Friedman, D. Geiger, M. Goldszmidt, Bayesian network classifiers, Mach. Learn. 29 (1997) 131–163.

[10] Dasarathy, B. V., Nearest Neighbor (NN) Norms,NN Pattern Classification Techniques. IEEE Computer Society Press, 1990.

[11] Wettschereck, D., Dietterich, T. G. “An Experimental Comparison of the Nearest Neighbor and Nearesthyperrectangle Algorithms,” Machine Learning, 9: 5-28, 1995.

[12] Platt J C. Fast Training of Support Vector Machines Using Sequential Minimal Optimization [M]. Advances in Kernel Methods:Support Vector Machines (Edited by Scholkopf B,Burges C,Smola A)[M]. Cambridge MA: MIT Press, 185-208, 1998.

[13] C. Burges, “A tutorial on support vector machines for pattern recognition”, Data Mining and Knowledge Discovery, vol. 2, 1998.

[14] Y. Yang and X. Liu, “A Re-Examination of Text Categorization Methods,” Proc. SIGIR ’99, pp. 42-49, 1999.

[15]T. Joachims, “Text Categorization with Support Vector Machines: Learning with Many Relevant Features,” Proc. European Conf. Machine Learning, pp. 137-142, 1998.

[16] Man Lan, Chew Lim Tan, Jian Su, and Yue Lu, “Supervised and Traditional Term Weighting Methods for Automatic Text Categorization”, Ieee Transactions on Pattern Analysis and Machine Intelligence, Vol. 31, No. 4, April 2009.