Sentiment Analysis of English Tweets Using RapidMiner

Pragya Tripathi, Santosh Kr Vishwakarma, Ajay Lala
Computer Science & Engineering, GGITS, Jabalpur, India
Email : pragya181@gmail.com, santoshscholar@gmail.com, ajaylala@ggits.org

Abstract – **Social networking sites these days are great source of communication for internet users. So these are important source for understanding the emotions of people. In this paper, we use data mining techniques for the purpose of classification to perform sentiment analysis on the views people have shared in Twitter. We collect dataset, i.e. the tweets from twitter that are in natual language and apply text mining techniques – tokenization, stemming etc to convert them into useful form and then use it for building sentiment classifier that is able to predict happy, sad and neutral sentiments for a particular tweet. Rapid Miner tool is being used, that helps in building the classifier as well as able to apply it to the testing dataset. We are using two different classifiers and also compare their results in order to find which one gives better results.**

Keywords: Sentiment analysis, natural language, data mining, text mining, Twitter.

I. INTRODUCTION

Sentiment analysis refers to the application of natural language processing, computational linguistics, and text analytics to identify and extract subjective information from the source materials. Sentiment analysis aims to determine the attitude of a speaker or a writer towards any topic or incident. It is the computational study of people's opinions, appraisals, attitudes, and emotions toward entities, individuals, issues, events, topics and their attributes. For this, there is need for data mining as well as text mining techniques. Data mining, also called knowledge discovery in Databases, is the process of discovering interesting and useful patterns and relationships in large volumes of data. The field combines tools from statistics and artificial intelligence (such as neural networks and machine learning) with database management to analyze large digital collections, known as data sets. It is the process of analyzing data from different perspectives and summarizing it into useful information. It employs data pre-processing, data analysis, and data interpretation processes in the course of data analysis. Data mining uses sophisticated mathematical algorithms[15]. Data classification is the process of organizing data into categories for its most effective and efficient use. Classification is a data mining function that assigns items in a collection to target categories or classes. The goal of classification is to accurately predict the target class for each case in the data. An algorithm that implements classification, especially in a concrete implementation, is known as a classifier. The term "classifier" sometimes also refers to the mathematical function, implemented by a classification algorithm that maps input data to a category[15].

Text mining is the analysis of data contained in natural language text. It is used to process unstructured (textual) information, extract meaningful numeric indices from the text. Generally some information retrieval methods, or natural language processing or some pre processing of text is done in order to make it useful for applying data mining(statistical and machine learning) algorithms. It is most beneficial when -

1) Summarizing documents
2) Extracting concepts from text
3) Indexing text for use in predictive analytics[10]

In this work, we are using two different classifiers to extract the sentiments of tweets people are sharing in twitter and classify them broadly into 3 categories – happy, sad and neutral. And also compare the precision and recall of each classifier and then also find out which classifier gives the best result in terms of better precision and recall ratios and accuracy.

II. RELATED WORK

A lot of work has been carried out in the field of sentiment analysis for the live data from the users in order to extract the sentiments of common people towards any topic, trend, products etc. The studies mainly focus on extracting useful information from the natural language of users and process it to get the real sentiments. It has gathered interest with the ever growing use of internet by people to share their opinions. Osaimi and Badruddin[1] have worked on the sentiment analysis of the tweets in Arabic language. They build different classifiers by training them with proper dataset i.e. processed tweets in Arabic language and then analyized the accuracy of these classifiers in order to predict the correct sentiments. Pak and Paroubek [2] used the twitter data as corpus to perform linguistic analysis and build a classifier that is highly efficient. A very broad overview of the existing work was presented by Pang and Lee[3]. In their survey, the authors describe existing techniques and approaches for an opinion-oriented information retrieval. Albert Bifet and Eibe Frank[4] have discussed the challenges that Twitter data streams pose, focusing on classification problems, and then consider these streams for opinion mining and sentiment analysis.

978-1-5090-0076-0/15 $31.00 © 2015 IEEE
DOI 10.1109/CICN.2015.137

III. METHODOLOGY

In this work, the tool that is used is Rapid Miner 5.3[16]. Rapid Miner is a software platform developed by the company of the same name that provides an integrated environment for machine learning, data mining, text mining, predictive analytics and business analytics. This is a sophisticated offering with over 1500 drag-and-drop operators with the help of which maximum data mining operations can be performed easily and quickly. For our work, we will the use operators of text mining, classification, validations etc. For converting natural language text useful for data mining we use text processing techniques- Tokenization that splits the text of a document into a sequence of tokens. Transform Cases that transforms all characters in a document to either lower case or upper case, respectively. Filter stop words which filters English stop words from a document by removing every token which equals a stop word from the built-in stop word list. Stem(Porter) stems English words using the Porter stemming algorithm applying an iterative, rule-based replacement of word suffixes intending to reduce the length of the words until a minimum length is reached. For the classification purpose, we use two most popular classifiers – Naive Bayes classifier and K-NN. A Naive Bayes classifier is a simple probabilistic classifier based on applying Bayes theorem (from Bayesian statistics) with strong (naive) independence assumptions. A naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature. Naive Bayes classifiers can handle an arbitrary number of independent variables, whether continuous or categorical. Given a set of variables, X = {x1, x2, x3..., xd}, we want to construct the posterior probability for the event Cj among a set of possible outcomes C = {c1, c2, c3..., cd}. In a more familiar language, X is the predictors and C is the set of categorical levels present in the dependent variable. Using Bayes' rule:

$$p\left(C_j \middle| x_1, x_2, \ldots, x_d\right) \propto p(x_1, x_2, \ldots, x_d)\middle| C_j\big)\, p(C_j) \quad - (1)$$

Where p(Cj | x1, x2, x3..., xd) is the posterior probability of class membership, i.e., the probability that X belongs to Cj. Since Naive Bayes assumes that the conditional probabilities of the independent variables are statistically independent we can decompose the likelihood of a product of terms:

$$p(X \mid C_j) \ \propto \ \prod_{k=1}^{d} p(x_k \mid C_j) \quad - (2)$$

And rewrite the posterior as:

$$p\left(C_j \mid X\right) \ \propto p(C_j) \ \prod_{k=1}^{d} p(x_k \mid C_j) \quad - (3)$$

Using Bayes' rule above, we label a new case X with a class level Cj that achieves the highest posterior probability[10].

K-Nearest Neighbor makes predictions based on the outcome of the *K* neighbors closest to that point. Therefore, to make predictions with *KNN*, we need to define a metric for measuring the distance between the query point and cases from the examples sample. One of the most popular choices to measure this distance is known as Euclidean.

$$D(x,p) = \sqrt{(x-p)^2} \quad - (4)$$

Where x and p are the query point and a case of the examples sample, respectively.

Since *KNN* predictions are based on the intuitive assumption that objects close in distance are potentially similar, it makes good sense to discriminate between the *K* nearest neighbors when making predictions. Let the closest points among the *K* nearest neighbors have more say in affecting the outcome of the query point. This can be achieved by introducing a set of weights *W*, one for each nearest neighbor, defined by the relative closeness of each neighbor with respect to the query point.

$$W(x,p_i) = \frac{\exp(-D(x,p_i))}{\sum_{i=1}^{k} \exp(-D(x,p_i))} \quad - (5)$$

Where $D(x, p_i)$ is the distance between the query point x and the ith case p_i of the example sample. The weights defined in this manner above will satisfy:

$$\sum_{i=1}^{k} W(x_0, x_i) = 1 \quad - (6)$$

Thus, for classification problems, the maximum of y is taken for each class variables.

$$\max\left(y = \sum_{i=1}^{k} W(x_0, x_i) y_i\right) \quad - (7)$$

Figure 1 shows the flow of main process. Process documents from files operator is used for reading text data available in any document file. Validation operator is used for providing training and applying different data mining algorithms in any process.

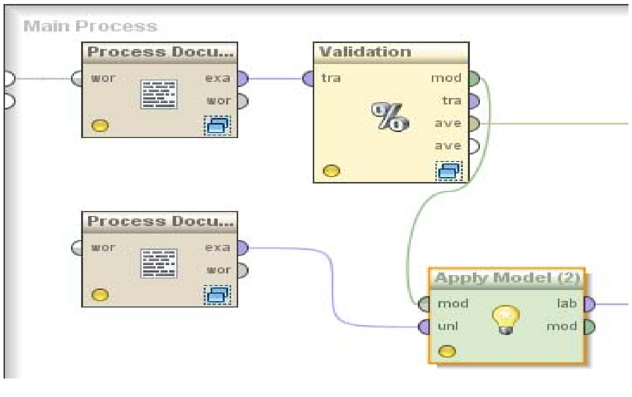


Figure 1: Main Process

Figure 2 shows the text mining operators used for processing the text files before applying for training and testing. Tokenize, filter tokens, transform case filter stop words and stemming operators are used to perform text mining related operations on the training and testing dataset. It is important to pre process the dataset because data available in natural language form cannot be directly used with data mining techniques. We process the text files so that we can use them for training the classifier and then these classifiers can predict labels for testing dataset.

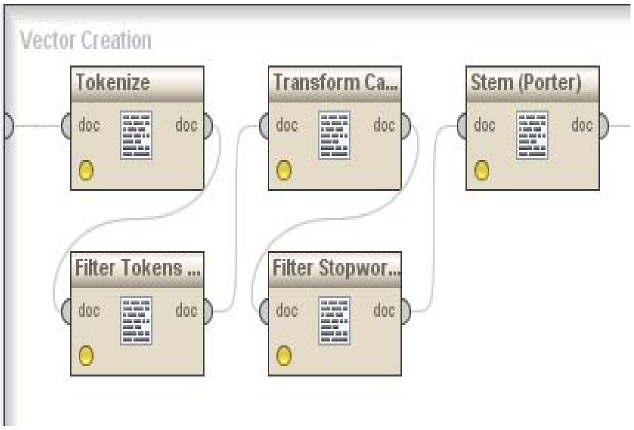


Figure 2: Text mining operators.

Figure 3 and figure 4 shows the sub processes within the X Validation operator. Naive Bayes classifier operator and K-NN classifier operator are being used respectively.

Apply Model is first trained on an Example Set; information related to the Example Set is learnt by the model. Then that model can be applied on another Example Set usually for prediction. It is compulsory that both Example Sets should have exactly the same number, order, type and role of attributes.Performance operator is used for performance evaluation of only classification tasks. For evaluating the statistical performance of a classification model the data set should be labelled i.e. it should have an attribute with *label* role and an attribute with *prediction* role. The *label* attribute stores the actual observed values, whereas the *prediction* attribute stores the values of *label* predicted by the classification model under discussion.

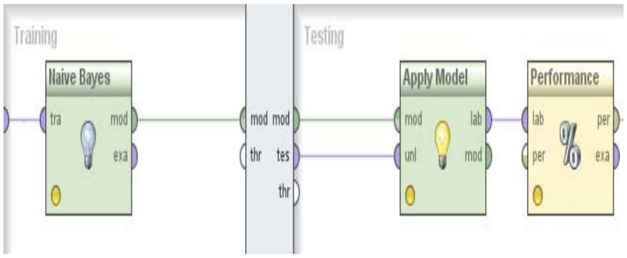


Figure 3: Naive Bayes Classifier

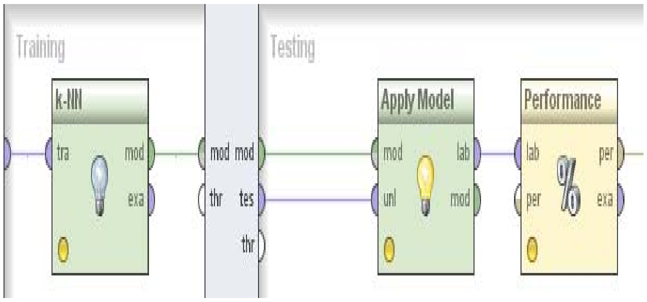


Figure 4: K-NN Classifier

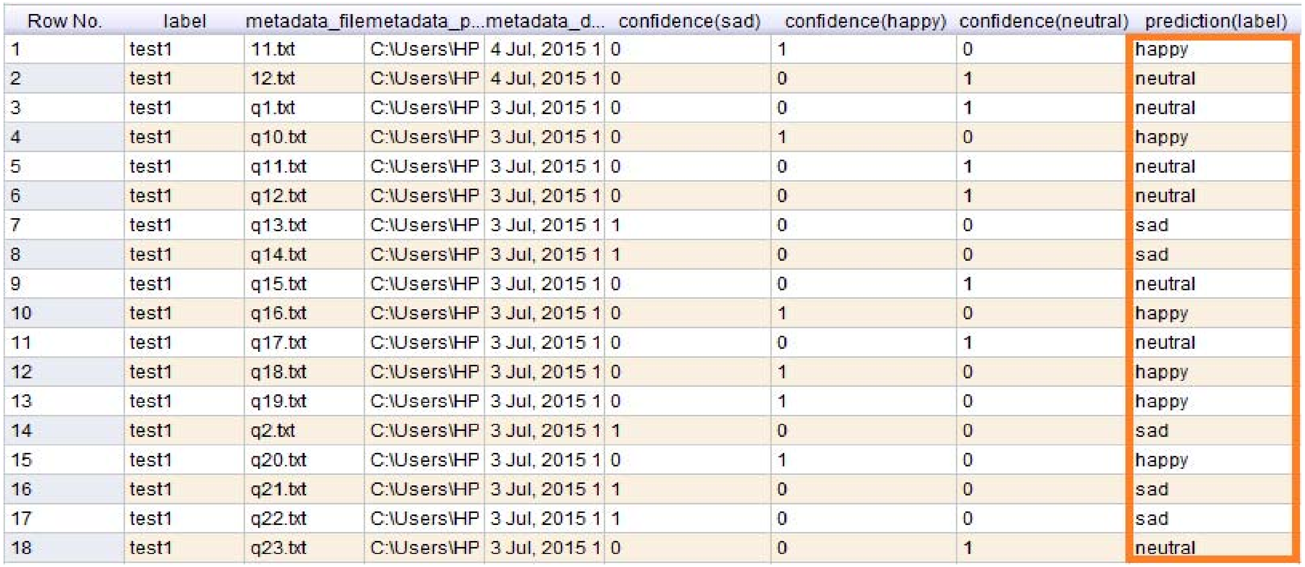
IV. EXPERIMENTS AND PERFORMANCE ANALYSIS

Dataset that is used in this work, are tweets are collected from the twitter.com[17] website which are in natural language. They are processed with the text mining operators like tokenization, filtering stopwords, stemming etc available in Rapid Miner before applying to the classifiers for training and testing. For providing training, we collected tweets and manually classified them into 3 types of class labels– happy, sad, neutral. These class labels will be used to train the classifier and then based on this learning predict the label of the testing dataset of the tweets. Table 1 shows the examples.

Label	Tweet
Happy	"we were happy because he qualified!"
Sad	"I'm real life sad right now & pissed & doubtful & just all over the place"
Netural	"I'm shy at first but I do the stupidest random things when I get comfortable with someone."

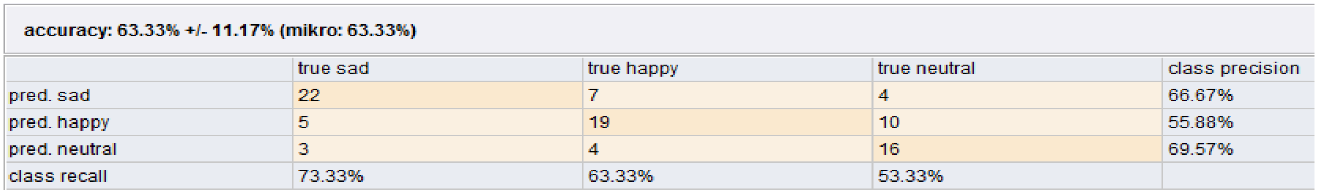
Table 1- Examples of tweets in the labels.

We collect a different dataset of tweets in the text files and provide them for the testing. Based on the learning, provided to the classifier during training, the labels for files in testing dataset are classified into one of the predefined labels - happy, sad, and neutral. We are using two different classifiers to predict the labels for testing dataset files. The results of predictions done by classifiers are shown in the figures below.The highlighted column prediction gives the information about the label predicted for the files present in the testing dataset.



Row No.	label	metadata_file	metadata_p...	metadata_d...	confidence(sad)	confidence(happy)	confidence(neutral)	prediction(label)
1	test1	11.txt	C:\Users\HP	4 Jul, 2015 1	0	1	0	happy
2	test1	12.txt	C:\Users\HP	4 Jul, 2015 1	0	0	1	neutral
3	test1	q1.txt	C:\Users\HP	3 Jul, 2015 1	0	0	1	neutral
4	test1	q10.txt	C:\Users\HP	3 Jul, 2015 1	0	1	0	happy
5	test1	q11.txt	C:\Users\HP	3 Jul, 2015 1	0	0	1	neutral
6	test1	q12.txt	C:\Users\HP	3 Jul, 2015 1	0	0	1	neutral
7	test1	q13.txt	C:\Users\HP	3 Jul, 2015 1	1	0	0	sad
8	test1	q14.txt	C:\Users\HP	3 Jul, 2015 1	1	0	0	sad
9	test1	q15.txt	C:\Users\HP	3 Jul, 2015 1	0	0	1	neutral
10	test1	q16.txt	C:\Users\HP	3 Jul, 2015 1	0	1	0	happy
11	test1	q17.txt	C:\Users\HP	3 Jul, 2015 1	0	0	1	neutral
12	test1	q18.txt	C:\Users\HP	3 Jul, 2015 1	0	1	0	happy
13	test1	q19.txt	C:\Users\HP	3 Jul, 2015 1	0	1	0	happy
14	test1	q2.txt	C:\Users\HP	3 Jul, 2015 1	1	0	0	sad
15	test1	q20.txt	C:\Users\HP	3 Jul, 2015 1	0	1	0	happy
16	test1	q21.txt	C:\Users\HP	3 Jul, 2015 1	1	0	0	sad
17	test1	q22.txt	C:\Users\HP	3 Jul, 2015 1	1	0	0	sad
18	test1	q23.txt	C:\Users\HP	3 Jul, 2015 1	0	0	1	neutral

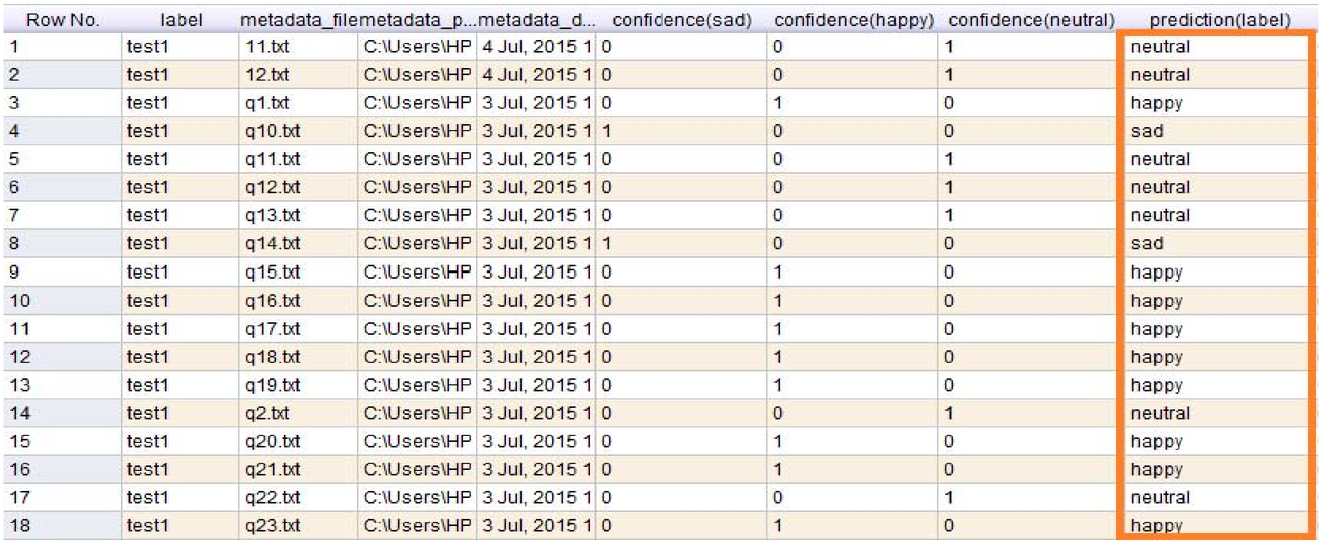
Figure 5: Results- Classification of tweets provided in the testing dataset using Naive Bayes Classifier



accuracy: 63.33% +/- 11.17% (mikro: 63.33%)

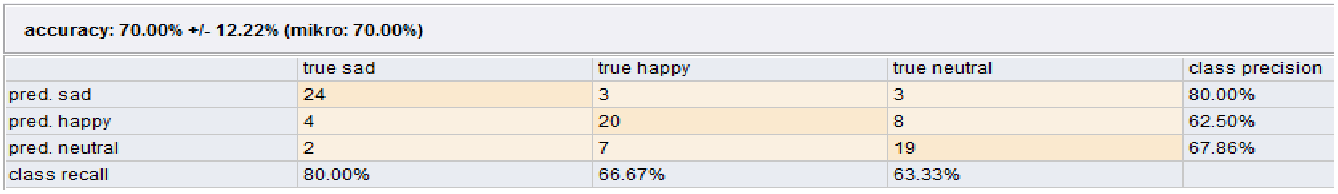
	true sad	true happy	true neutral	class precision
pred. sad	22	7	4	66.67%
pred. happy	5	19	10	55.88%
pred. neutral	3	4	16	69.57%
class recall	73.33%	63.33%	53.33%	

Figure 6: Precision and Recall ratio for the classification



Row No.	label	metadata_file	metadata_p...	metadata_d...	confidence(sad)	confidence(happy)	confidence(neutral)	prediction(label)
1	test1	11.txt	C:\Users\HP	4 Jul, 2015 1	0	0	1	neutral
2	test1	12.txt	C:\Users\HP	4 Jul, 2015 1	0	0	1	neutral
3	test1	q1.txt	C:\Users\HP	3 Jul, 2015 1	0	1	0	happy
4	test1	q10.txt	C:\Users\HP	3 Jul, 2015 1	1	0	0	sad
5	test1	q11.txt	C:\Users\HP	3 Jul, 2015 1	0	0	1	neutral
6	test1	q12.txt	C:\Users\HP	3 Jul, 2015 1	0	0	1	neutral
7	test1	q13.txt	C:\Users\HP	3 Jul, 2015 1	0	0	1	neutral
8	test1	q14.txt	C:\Users\HP	3 Jul, 2015 1	1	0	0	sad
9	test1	q15.txt	C:\Users\HP	3 Jul, 2015 1	0	1	0	happy
10	test1	q16.txt	C:\Users\HP	3 Jul, 2015 1	0	1	0	happy
11	test1	q17.txt	C:\Users\HP	3 Jul, 2015 1	0	1	0	happy
12	test1	q18.txt	C:\Users\HP	3 Jul, 2015 1	0	1	0	happy
13	test1	q19.txt	C:\Users\HP	3 Jul, 2015 1	0	1	0	happy
14	test1	q2.txt	C:\Users\HP	3 Jul, 2015 1	0	0	1	neutral
15	test1	q20.txt	C:\Users\HP	3 Jul, 2015 1	0	1	0	happy
16	test1	q21.txt	C:\Users\HP	3 Jul, 2015 1	0	1	0	happy
17	test1	q22.txt	C:\Users\HP	3 Jul, 2015 1	0	0	1	neutral
18	test1	q23.txt	C:\Users\HP	3 Jul, 2015 1	0	1	0	happy

Figure 7: Results- Classification of tweets provided in the testing dataset using K-NN Classifier



accuracy: 70.00% +/- 12.22% (mikro: 70.00%)

	true sad	true happy	true neutral	class precision
pred. sad	24	3	3	80.00%
pred. happy	4	20	8	62.50%
pred. neutral	2	7	19	67.86%
class recall	80.00%	66.67%	63.33%	

Figure 8: Precision and Recall ratio for the classification

Figure 5 shows the classification of tweets using the Naive Bayes classifier in the testing dataset into the labels, i.e. happy, sad, and neutral, according to learning provided to the Naive Bayes classifier during the time of training in the highlighted column(prediction). Figure 6 shows the accuracy, precision and recall ratios of the Naive Bayes classifier for the prediction done on the testing dataset. Figure 7 shows the classification of tweets using the K-NN classifier into the labels, i.e. happy, sad, and neutral, according to learning provided to the K-NN classifier during the time of training in the highlighted column(prediction) . Figure 8 shows the accuracy, precision and recall ratios of the K-NN classifier for the prediction done on the testing dataset.

V. CONCLUSION

In this paper, sentiment classification of tweets is done into different labels with the help of data mining and text mining techniques. We used two classifiers – Naive Bayes and K-NN. These classifiers are trained with the training dataset and then testing dataset is applied. Both the classifiers predicted the labels for testing dataset. As we compare the results of both we can see that in the current setup, K-NN classifier gives more accurate prediction, so we can assume it is better classifer in given situation.

In future we aim to increase the size of the testing dataset and look for the complex issues related to it. The number of labels can also be increased or changed based on the needs. We aim to use emticons as well for classification in future. This scheme canbe applied to other Indian languages (Hindi, Marathi, Gujarati, and Tamil etc). The results of this research will be synced with the actual mood detection and customer feedback system.

VI. REFERENCES

[1] Salha al Osaimi, Khan Muhammad BAdruddin, Dept of Information Systems, Imam Muhammad ibn Saud Islamic University, KSA. “Sentiment Analysis of Arabic tweets Using RapidMiner.”

[2] Alexander Pak, Patrick Paroubek from Universit´e de Paris-Sud, Laboratoire LIMSI-CNRS, Bˆatiment 508,F-91405 OrsayCedex, France, “Twitter as a Corpus for Sentiment Analysis and Opinion Mining.”

[3] Albert Bifet and Eibe Frank from University of Waikato, Hamilton, New Zealand, “Sentiment Knowledge Discovery in Twitter Streaming Data.”

[4] Bo Pang and Lillian Lee from Yahoo! Research, 701 First Avenue, Sunnyvale, CA 94089, USA,Computer Science Department, Cornell University, Ithaca, NY 14853, USA. “Opinion Mining and Sentiment Analysis”.

[5] Dmitry Davidov, Oren Tsur, Ari Rappoport ICNC Institute of Computer Science, The Hebrew University, enhanced Sentiment Learning Using Twitter Hashtags and Smileys.”

[6] Efthymios Kouloumpis I-sieve Technologies, Athens, Greece, Theresa Wilson HLT Center of Excellence, Johns Hopkins University, Baltimore, MD, USA, Johanna Moore School of Informatics, University of Edinburgh, Edinburgh, UK, “Twitter Sentiment Analysis: The Good the Bad and the OMG!”.

[7] Soo-Min Kim, Eduard Hovy Information Sciences Institute, University of Southern California 4676 Admiralty Way Marina del Rey, CA 90292-6695 “Determining the Sentiment of Opinions”.

[8] Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. Found. Trends Inf. Retr., 2(1-2):1–135. 2008.

[9] Balahur, A., Steinberger, R. Rethinking Opinion Mining in News: from Theory to Practice and Back. In Proceedings of the 1st Workshop on Opinion Mining and Sentiment Analysis, Satellite to CAEPIA 2009.

[10] Umajancy.S, Dr. Antony Selvadoss Thanamani , “AN ANALYSIS ON TEXT MINING -TEXT RETRIEVAL AND TEXT EXTRACTION” International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 8, August 2013.

[11] Handbook of Natural Language Processing, Second Edition, edited by Nitin Indurkhya, Fred J. Damerau.

[12] Mining the Social Web: Data Mining Facebook, Twitter, LinkedIn, Google+ By Matthew A. Russell

[13] Data Mining Methods for the Content Analyst. By Kalev Leetaru.

[14] Data Mining Techniques, Tenth Edition by Arun K Pujari.

[15] Data Mining Concepts and techniques, Third Edition by Kamber

[16] RapidMiner.com

[17] Twitter.com