A Study of Product Trend Analysis of Review Datasets using Naive Bayes’, K-NN and SVM Classifiers

Sanjay Chakraborty, Shailesh Kumar, Soudip Paul and Animesh Kairi

Abstract
Trend analysis of datasets is one kind of text mining procedure which is a part of natural language processing. Product trend analysis is basically the method of analyzing the reviews given by the customer to a particular product. In this paper, we have used various classifier’s algorithms (such as, Naive Bayes, K-NN and SVM) to determine the positivity or negativity of the review datasets on recent market trends. It proposes that the trend of a particular product through which we can predict whether it will continue in the market or not. The whole database of this work is a collection of user reviews (such as, comments, testimonials, messages etc.) from various social sites. This market trends analysis has a huge advantage in terms of monetization and profits. This approach also shows the comparison among these classifiers with respect to the feedbacks of market data analysis. Finally, these comparisons are also included movie reviews dataset. It shows that SVM classifier performs better analysis in terms of precision, recall and accuracy parameters on the above datasets compare to the widely used K-NN and Naïve Bayes machine learning classifiers.

Keywords
K-NN;
Naïve Bayes,
Opinion Mining Classifier,
Product Trend Analysis.
SVM,
Text mining,

I. INTRODUCTION
In this modern era people are always looking for the feedbacks of others before using or experiencing any product. Sentiment analysis or opinion mining is one of the significant applications of the natural language processing or computer linguistic to identify the sense or the mood of the material. Clustering-based sentiment analysis is a novel approach for analyzing opinions expressed in reviews, comments or blogs. Various clustering algorithms (K-means [16][17][18]. DBSCAN[15]. Hierarchical etc.) are widely used for this type of opinion mining. In this work, materials are the customer’s reviews / opinions / comments given in the various online stores to the products which have been bought and used. According to the sense or mood of the reviews or comments we can send them to two different polarities. If the sense of a review is positive then it will be positive polarized or if that has a sense of negativity then it will be negatively polarized. As for example,
1) If a review is like “I am very satisfied with the quality of the product. It is also profitable in terms of economical point of view. I am very satisfied.” This sentence expresses someone’s satisfactory mood for a particular product, so it is a positive review.
2) Another review likes, “I have purchased this product two months back. But now it stops working. I am totally disappointed. Bad product.” this sentence express someone’s unsatisfactory mood, so it is a negative review.

Traditional text mining deals with the facts whereas sentiment analysis uses to analyze the mood of the sentences. Now a day’s market analysis becomes a popular and necessary process for all the online and offline stores to a get probabilistic result for marketing of the products according to the customer’s feedback. There are various machine learning algorithms which are used for classification, analysis and prediction of useful data from a large dataset. This paper mainly focuses on three widely used supervised machine learning algorithms (Naïve-Bayes, K-Nearest Neighbour and Support vector machine) for product trend analysis [12]. Here, we are using three factors accuracies, precisions, recall values with the applications of above classification algorithms for analysis the
actual product trends in the market. In pattern recognition and information retrieval with classification approaches, precision is a fraction of the collected relevant instances and recall is a fraction of relevant instances that are retrieved. Precision and Recall can be obtained from Eq. (1) and Eq. (2).

\[
\text{Precision} = \frac{TP}{TP+FP} \quad (1) \\
\text{Recall} = \frac{TP}{TP+FN} \quad (2)
\]

Where, True positive (TP) means identical with hit, True negative (TN) means identical with correct rejection and False positive (FP) means identical with false alarm and False negative (FN) means identical with miss.

The challenges in sentiment analysis are the words which can be used in dual senses. For that kind of word, it is very difficult to justify the real sense as whether that is used as a positive or a negative mood using any algorithm. This may give some inaccurate results. Another problem is the degree of intensity of a word used in a sentence. For example good and very good both are positive polarized words but very good have a more intense sense than good. It is also a challenge to determine by the algorithms. Though, we have used the scaling of words where the concept of very positive or very negative are introduced. Using different classification algorithms on the same review data set, we will measure the accuracy of the results and compare them to achieve the best result. The rest of the paper is organized as follows, Section II describes the related works, section III explains the proposed work with suitable examples (data sources, algorithms, methodologies used in our study), and section IV describes all the experimental views and a detailed analysis of them and at last section V deals with the conclusion and future work of this approach.

II. RELATED WORK

There are several sentiment analysis work done on different topics. Here are some of the previous works are explained below,

In paper [1], ‘Naïve Bayes’ and neural network classifiers are combined together for movie reviews classification. In this work, the combined classification model classified the movie reviews into positive and negative polarities of sentiments with the help of WordStat keywords dictionary. The dictionary is classified into set of positive and negative words where the occurrence of each word is used as a feature for training a classifier. After that, the processed data is passed into the above two combined classifiers for classification and based on the input testing file, the output is produced and evaluated in the form of confusion matrix.

Some of the approaches [2] classify the sentiments into different groups using the necessary keywords based selection technique. Those necessary keywords are basically the adjectives ( sometime used well-known Affine list) to indicate various sentiment. In most cases, an updated indicators list is prepared manually and used them for sentiment analysis. The paper [3] has done online movie reviews using sentiment analysis approaches. For this work, it takes the help of three supervised machine learning approaches SVM, Naïve Bayes and K-NN.

In the paper [4], twitter data is classified using Naïve Bayes classifier. In this paper, twitter is considered as micro blogging website which contains smaller data compare to large blogs. Twitter search APIs is used to extract useful twitter data for classification based on its contents. In the article [5], document and sentence level classification is performed from different product destinations into positive and negative categories. Then the overall positive or negative score for the text is evaluated and perform comparison between the positive and negative documents based on the high density of the positive or negative words present in the documents. In paper [6], a work on sentiment classification has done using single pass clustering for K-Nearest Neighbor algorithm. A very similar work on hotel customer reviews and movie reviews are analyzed using the well-known K-NN and Naïve Bayes classification algorithms in paper [12].

In the article [9], an improved version of K-NN text classification algorithm using clustering is introduced. Most of the discussed work is working on a huge number of training samples and has a high degree of calculation complexity. In this innovative approach, first the training sets are compressed and unnecessary border line samples are removed.
Secondly, the training samples are clustered using K-means clustering and all cluster centers are taken as the updated training samples. Finally, this less number of modified samples is used as input to the K-NN text classification algorithm.

In papers [10][14], the enhancement of various newly proposed sentiments analysis algorithms and their applications are investigated and presented in a comprehensive manner. Besides that, the related fields of sentiment analysis algorithms like, transfer learning, emotion detection, and building resources are also discussed in this survey.

III. PROPOSED WORK

A) Working Data Set
For this study first we use a product review data set. All the reviews are collected from online shopping site like amazon. The source data set is the collection of 10000 customer reviews on mobile products where both the positive and negative reviews are mixed with each other.

B) Methodology
The aim of our study is to analyze the various customer reviews and also take decision about the future analysis using the described classifiers.

i. Firstly data collection was done in order to make a good dataset of product reviews. We have collected the raw data from the internet source http://www.jmcauley.ucsd.edu/data/amazon/.

ii. Collected raw data was converted to a text document and saved as read.csv file.

iii. From that raw data set for each review we have stripped of white spaces and punctuation.

iv. Also we have removed all the numbers and the stop words from those reviews in order to get the review text only. Stemming is applied to get the desired data set. A pseudo code (Fig 1) is given below by which we got the desired.

v. Stemming is applied to get the desired data set. A pseudo code is given below by which we got the desired review data set from the raw data set.

vi. Different Classifiers are used to get the prediction based on the product reviews.

vii. Prediction Output is compared in order to determine the best method to predict the product trend.

The process of sentiment analysis is given in a graphical view (Fig 2).

1) Naïve Bayes Classifier
In machine learning, ‘Naïve Bayes’ classifiers are a family of simple probabilistic classifiers which uses the concept of Bayes theorem with naïve independence assumptions between the features. It is one kind of supervised classification paradigm. It was used to determine the text using the occurrence of words and classify that to different category. It uses a small amount of training dataset to evaluate the parameters required for the classification. This conditional probabilistic model works satisfactorily despite its strong assumption. In Naïve Bayes, it first calculates the joint probabilities of words and categories. Then it is used to find the probabilities of the desired text document.

### Code used to convert the raw dataset into workable data set
```r
data8<- read.csv(file="C:\Users\Shailesh\Desktop\mixed4.txt")
k=1
pos=0
neg=0
for(i in data$V22){
j=substring(i,1,3)
j=as.numeric(j)
if(j>=3){
data8[k,2]="pos"
    pos=pos+1
}
else{
data8[k,2]="neg"
   neg=neg+1
}
k=k+1;
```

Fig 1. Code used to convert the raw dataset into workable data set

![Data Collection](Fig1.png)
The Bayes theory for conditional probability states that for a given data value k and a class CL (Eq 3),

\[ P(CL/k) = \frac{P(k/CL) P(CL)}{P(k)} \quad (3) \]

Also by taking the assumption that for a data point \( k = \{k_1,k_2,\ldots,k_i\} \), the probability of each of its attribute occurring in a given class is independent and then the probability of \( k \) can be estimated as Eq (4),

\[ P(CL/k) = P(CL) \prod P(k/CL) \quad (4) \]

The function that is to determine the results for the test is given below:

```r
classifier = naiveBayes(mat[1:5000,], as.factor(data[1:5000,2]))
predicted = predict(classifier, mat[5001:6000,])
```

**Algorithm**

Input: a required document \( d \)

A predefined set of classes \( CL = \{c_1, c_2, \ldots, c_l\} \)

Output: a predicted class \( c_l \in CL \)

**Work Flow:**

i. Firstly 10000 reviews of customer about different product were taken from a database of about more than 1 million reviews collected from amazon customer review database.

ii. Reviews were classified as positive and negative and kept them both in a single file.

iii. The positive and negative reviews of a sentence was given tag as pos or neg by appending with them accordingly.

iv. As per the standard, 70% of the data samples were considered as the training dataset and rest were taken as the test dataset.

v. The classifier was trained using the set of training data.

vi. Labeled sentences were taken and compare with training dataset using them as test dataset.

**2) K-Nearest Neighbor Classifier**

K-NN is a type of instance based learning where it saves all available features and classifies new features based on a similarity measure [8]. The similarity measure generally uses various distance based functions. K-NN is the simplest method which is used to classify data of which category that belongs to. For classification an object is classified by majority poles of its k-nearest neighbors. The test samples are classified using the distance with the smallest value corresponds to the sample in the training set. This approach is a simple but very powerful algorithm to determine the sentiment analysis. It is very powerful tool because it does not acquire anything except the distance measure between two nearest instances [6][7]. But the drawback of this simple K-NN approach is the lack of robustness. The function that is used to determine the results for the test is given below:

\[ K-NNresult <= K-NN(train,test,class,k=20) \]

**Work Flow:**

i. Firstly 10000 reviews of customer about different product were taken from a database of about more than 1 million reviews collected from amazon customer review database.

ii. Reviews were classified as positive and negative and kept them both in a single file.

iii. Tagging as pos or neg according to the positive and negative reviews of the input sentence.

iv. As per the standard, 70% of the data samples were considered as the training dataset and rest were taken as the test dataset.

v. Then the score of each word are calculated in the training dataset and store them in an empty dataset with their scores.

vi. For each data set if a word matched with the word set, then its score is added to review score.

vii. Evaluate the matrices according to the scores.

**3) Support Vector Machines Classifier**

Support Vector Machines (SVMs) are supervised learning methods widely used for classification. It is a theoretical model which guarantees about its performance and it does not affected by local minima and suffered from the curse of dimensionality. Non-overlapping partitions are not generated in SVM classification model. SVMs maximize the margin around the separating hyperplane. The decision function is fully specified by a subset of training samples, the support vectors [14]. Maximum margin among the possible hyperplanes can be chosen by finding the weight.
vector w and the bias b. The function that is to determine the results for the test is given below:

\[
SVMresult <= SVM(dataset[1:5000,2]):-,train)
Predicted = predict(SVMresult,dataset[5001:6000])
\]

**Work Flow:**
i. Firstly 10000 reviews of customer about different product were taken from a database of about more than 1 million reviews collected from amazon customer review database.
ii. Randomly we divided 70% of the instances as training, and another 30% of instances as testing.
iii. Next the parameters (C - cost, gamma - gamma) are chosen using tune function. Where, 0.000001<gamma<=0.1 and 0.1<C<=10.
iv. Best parameters value from above step was chosen as parameter to train the SVM model.
v. The SVM classifier was trained using the set of training data.
vi. To predict the test set classes, the model is evaluated again with the help of prediction function.

**IV. EXPERIMENTAL RESULTS**

In this experiment, accuracy, precision and recall parameters are mainly used for performance evaluation. Accuracy, precision and recall are discussed in section 1 in detail. All the experiments have been done in the R platform (version 3.1.2). We have installed the e1071, tm, SnowballC, RTextTools packages in R to do the trend analysis operations. The machine configuration is also mentioned below,

i. Intel core i3 3rd gen processor
ii. 500 GB HDD
iii. 4 GB RAM
iv On Windows 8.1 operating system.

Table 1 shows the actual experimental result and other results are shown in the tables below. Rewrite the definitions of accuracy, precision and recall from section 1.

**Accuracy** = (True Positive + True Negative) / Total no. of rows in test data set

**Precision** = True Positive / (True Positive + False Positive)

**Recall** = True Positive / (True Positive + False Positive)

<table>
<thead>
<tr>
<th>Experiment No</th>
<th>Number of Reviews</th>
<th>Native Bayes</th>
<th>K-NN</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>40.19</td>
<td>43.21</td>
<td>49.22</td>
</tr>
<tr>
<td>2</td>
<td>200</td>
<td>41.73</td>
<td>37.49</td>
<td>48.61</td>
</tr>
<tr>
<td>3</td>
<td>500</td>
<td>40.81</td>
<td>40.11</td>
<td>53.71</td>
</tr>
<tr>
<td>4</td>
<td>1000</td>
<td>41.69</td>
<td>40.83</td>
<td>55.17</td>
</tr>
<tr>
<td>5</td>
<td>2000</td>
<td>42.28</td>
<td>41.4</td>
<td>59.37</td>
</tr>
<tr>
<td>6</td>
<td>3000</td>
<td>47.44</td>
<td>44.52</td>
<td>61.82</td>
</tr>
<tr>
<td>7</td>
<td>4000</td>
<td>50.19</td>
<td>46.18</td>
<td>65.3</td>
</tr>
<tr>
<td>8</td>
<td>4500</td>
<td>52.38</td>
<td>50.51</td>
<td>70.36</td>
</tr>
</tbody>
</table>

However, we are evaluating the same parameters (recall, precision and accuracy) for movie review


208
dataset below [12]. In the below tables, we are included the SVM classifier evaluation for opinion mining of movie reviews (Table 3).

<table>
<thead>
<tr>
<th>Experiment No.</th>
<th>Number of Reviews</th>
<th>NAIVE BAYES</th>
<th>K-NN</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>42.81</td>
<td>44.53</td>
<td>49.29</td>
</tr>
<tr>
<td>2</td>
<td>200</td>
<td>41.64</td>
<td>41.31</td>
<td>48.52</td>
</tr>
<tr>
<td>3</td>
<td>500</td>
<td>41.83</td>
<td>42.47</td>
<td>53.19</td>
</tr>
<tr>
<td>4</td>
<td>1000</td>
<td>45.49</td>
<td>40.32</td>
<td>55.94</td>
</tr>
<tr>
<td>5</td>
<td>2000</td>
<td>47.7</td>
<td>44.69</td>
<td>59.85</td>
</tr>
<tr>
<td>6</td>
<td>3000</td>
<td>49.66</td>
<td>45.83</td>
<td>63.25</td>
</tr>
<tr>
<td>7</td>
<td>4000</td>
<td>50.24</td>
<td>46.72</td>
<td>67.80</td>
</tr>
<tr>
<td>8</td>
<td>4500</td>
<td>51.38</td>
<td>51.26</td>
<td>70.47</td>
</tr>
</tbody>
</table>

Diagrammatic representation of Precision in the test dataset is shown in Fig 4 and Diagrammatic representation of recall in the test result output shown in Fig 5 obtained from Table 5.

<table>
<thead>
<tr>
<th>Experiment No.</th>
<th>Number of Reviews</th>
<th>Native Bayes</th>
<th>K-NN</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>0.37</td>
<td>0.40</td>
<td>0.51</td>
</tr>
<tr>
<td>2</td>
<td>200</td>
<td>0.41</td>
<td>0.41</td>
<td>0.48</td>
</tr>
<tr>
<td>3</td>
<td>500</td>
<td>0.42</td>
<td>0.41</td>
<td>0.53</td>
</tr>
<tr>
<td>4</td>
<td>1000</td>
<td>0.43</td>
<td>0.39</td>
<td>0.54</td>
</tr>
<tr>
<td>5</td>
<td>2000</td>
<td>0.42</td>
<td>0.41</td>
<td>0.59</td>
</tr>
<tr>
<td>6</td>
<td>3000</td>
<td>0.47</td>
<td>0.43</td>
<td>0.66</td>
</tr>
<tr>
<td>7</td>
<td>4000</td>
<td>0.51</td>
<td>0.44</td>
<td>0.67</td>
</tr>
<tr>
<td>8</td>
<td>4500</td>
<td>0.57</td>
<td>0.46</td>
<td>0.71</td>
</tr>
</tbody>
</table>

The Fig 6 and Fig 7 show the diagrammatic recall (%) comparison of Naïve Bayes, K-NN and SVM algorithms on movie reviews test datasets for positive and negative corpus respectively. On movie reviews dataset, we have performed seven rounds of experiments.

Table 6. Recall (%) comparison for negative corpus in Movie Reviews test dataset

<table>
<thead>
<tr>
<th>Experiment No.</th>
<th>Number of Reviews</th>
<th>NAIVE BAYES</th>
<th>K-NN</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>69.24</td>
<td>39.25</td>
<td>68.76</td>
</tr>
<tr>
<td>2</td>
<td>500</td>
<td>68.79</td>
<td>53.86</td>
<td>70.89</td>
</tr>
<tr>
<td>3</td>
<td>1000</td>
<td>73.44</td>
<td>60.21</td>
<td>76.29</td>
</tr>
<tr>
<td>4</td>
<td>2000</td>
<td>81.02</td>
<td>65.03</td>
<td>80.91</td>
</tr>
<tr>
<td>5</td>
<td>3000</td>
<td>82.44</td>
<td>66.52</td>
<td>84.69</td>
</tr>
<tr>
<td>6</td>
<td>4000</td>
<td>83.34</td>
<td>66.25</td>
<td>85.37</td>
</tr>
<tr>
<td>7</td>
<td>4500</td>
<td>84.84</td>
<td>66.31</td>
<td>83.58</td>
</tr>
</tbody>
</table>

The Fig. 8 and Fig. 9 show the diagrammatic precision (%) comparison of Naïve Bayes, K-NN and SVM algorithms on movie reviews test datasets for positive and negative corpus respectively.

Table 7. Precision (%) comparison for positive corpus in Movie Reviews test dataset

<table>
<thead>
<tr>
<th>Experiment No.</th>
<th>Number of Reviews</th>
<th>NAIVE BAYES</th>
<th>K-NN</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>59.04</td>
<td>41.35</td>
<td>65.76</td>
</tr>
<tr>
<td>2</td>
<td>500</td>
<td>69.56</td>
<td>54.42</td>
<td>71.89</td>
</tr>
<tr>
<td>3</td>
<td>1000</td>
<td>73.64</td>
<td>58.18</td>
<td>77.29</td>
</tr>
<tr>
<td>4</td>
<td>2000</td>
<td>80.28</td>
<td>65.57</td>
<td>79.91</td>
</tr>
<tr>
<td>5</td>
<td>3000</td>
<td>81.64</td>
<td>67.03</td>
<td>84.69</td>
</tr>
<tr>
<td>6</td>
<td>4000</td>
<td>82.92</td>
<td>67.75</td>
<td>85.75</td>
</tr>
<tr>
<td>7</td>
<td>4500</td>
<td>84.09</td>
<td>68.14</td>
<td>82.58</td>
</tr>
</tbody>
</table>

Table 8. Precision (%) comparison for negative corpus in Movie Reviews test dataset

<table>
<thead>
<tr>
<th>Experiment No.</th>
<th>Number of Reviews</th>
<th>NAIVE BAYES</th>
<th>K-NN</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>55.43</td>
<td>38.12</td>
<td>58.26</td>
</tr>
<tr>
<td>2</td>
<td>500</td>
<td>70.59</td>
<td>57.25</td>
<td>72.89</td>
</tr>
<tr>
<td>3</td>
<td>1000</td>
<td>73.99</td>
<td>62.12</td>
<td>71.39</td>
</tr>
<tr>
<td>4</td>
<td>2000</td>
<td>78.09</td>
<td>65.73</td>
<td>81.91</td>
</tr>
<tr>
<td>5</td>
<td>3000</td>
<td>79.00</td>
<td>66.47</td>
<td>82.45</td>
</tr>
<tr>
<td>6</td>
<td>4000</td>
<td>81.33</td>
<td>66.62</td>
<td>80.20</td>
</tr>
<tr>
<td>7</td>
<td>4500</td>
<td>81.01</td>
<td>66.73</td>
<td>83.78</td>
</tr>
</tbody>
</table>

Table 9 shows number of correct classifications and incorrect classification using the accuracy results. Naïve Bayes’ algorithm generates more accurate samples than K-NN algorithm and SVM generates more than Naïve Bayes’. So, overall SVM classifier generates better results than naïve bayes’ and K-NN classifiers.

**Table 9.** Result of accuracies with maximum number of correct classifications on movie reviews dataset

<table>
<thead>
<tr>
<th>Total Reviews</th>
<th>Classifiers</th>
<th>Correct Sample</th>
<th>Incorrect Sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>1500</td>
<td>K-NN</td>
<td>1047</td>
<td>453</td>
</tr>
<tr>
<td></td>
<td>Naïve Bayes</td>
<td>1237</td>
<td>263</td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>1365</td>
<td>135</td>
</tr>
</tbody>
</table>

**Table 10.** Accuracy comparison on Movie reviews Dataset

<table>
<thead>
<tr>
<th>Experiment No.</th>
<th>Number of Reviews</th>
<th>NAIVE BAYES</th>
<th>K-NN</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>56.78</td>
<td>47.64</td>
<td>59.25</td>
</tr>
<tr>
<td>2</td>
<td>500</td>
<td>70.06</td>
<td>58.44</td>
<td>72.89</td>
</tr>
<tr>
<td>3</td>
<td>1000</td>
<td>73.81</td>
<td>61.48</td>
<td>73.29</td>
</tr>
<tr>
<td>4</td>
<td>2000</td>
<td>79.14</td>
<td>66.02</td>
<td>81.91</td>
</tr>
<tr>
<td>5</td>
<td>3000</td>
<td>80.27</td>
<td>68.58</td>
<td>82.36</td>
</tr>
<tr>
<td>6</td>
<td>4000</td>
<td>82.11</td>
<td>69.03</td>
<td>84.92</td>
</tr>
<tr>
<td>7</td>
<td>4500</td>
<td>82.43</td>
<td>69.81</td>
<td>82.87</td>
</tr>
</tbody>
</table>

**Fig 1.** Diagrammatic presentation of mean accuracies in the move review experiment

But SVM is giving better results on average than both Naïve Bayes and K-NN algorithm for our product review dataset. We have also shown how SVM gives better results on movie reviews dataset [12] compare to K-NN and Naïve Bayes algorithms. Thus we can conclude that SVM classifier can be used successfully for analyzing the customer reviews for products.

**Fig 11.** Comparison of Mean Accuracies in the test result of product reviews data

All these above analysis help us to predict the flow of products into the market which can increase the profit margin of retail organizations.

**Fig 12.** Comparison of Mean Accuracies on movie reviews data

The main comparison with the paper [12] lies that this work uses the most popular support vector machine (SVM) classification algorithm for analysis.


DOI: https://doi.org/10.24999/IJOAEM/02090047
and prediction purpose. In paper [12], only K-NN and Naive Bayes supervised machine learning algorithms are used to analyze the movie reviews and hotel reviews. Like our proposed analysis, we have also shown (Fig.7, Fig.8 and Fig.9) how SVM will give better result on movie review dataset in terms of recall, precision and accuracy. The overall mean accuracy on movie reviews dataset are compared and shown in Fig.10. It also shows a comparison of mean accuracies of product review and movie review datasets in Fig.11 and Fig.12 respectively. From the experimental results we can find that Naïve Bayes and K-NN are giving almost same results or sometimes Naïve Bayes is giving better result over K-NN algorithm.

V. CONCLUSION AND FUTURE WORK
The main focus of this work is to measure the performance of product trend analysis in terms of accuracy, precision and recall factors. In this paper, three very popular machine learning algorithms are used on the market product reviews dataset and finally compare the results of each classification. For future work we would try to make a more efficient sentiment analyzer taking the advantages of all the algorithms. Also we would try to add a crawler with the analyzer which can automatically add more reviews to its dataset to make the dataset updated with new records and sentiments in order to achieve more accurate results that can be used for good product trend analyzer in market analysis.

REFERENCES


Authors

Sanjay Chakraborty
Dept. of Computer Science and Engineering
Institute of Engineering and Management, Kolkata, India.
Email: sanjay.chakraborty@iemcal.com

Shailesh Kumar
Dept. of Computer Science and Engineering
Institute of Engineering and Management, Kolkata, India.
Email: shaileshrockstar5@gmail.com

Soudip Paul
Dept. of Computer Science and Engineering
Institute of Engineering and Management, Kolkata, India.
Email: soudip.pal@gmail.com

Animesh Kairi
Dept. of Information Technology
Institute of Engineering and Management, Kolkata, India.
Email: animesh.kairi@iemcal.com