

# Product review classification using LSA with improved feature weight model

S. Poomagal, R. Santhosh and B. Sanjay Prakash  
{ bsp.amcs@psgtech.ac.in }  
santhoshrajendran2001@gmail.com, sanjayprakashsivakasi@gmail.com

Department of Applied Mathematics and Computational Sciences,  
PSG College of Technology, Coimbatore.

## Abstract

Product review classification plays a vital role in understanding the likes and dislikes of users of the product. This analysis can be utilized to improve the quality of product as expected by users. In this paper, classification of reviews is performed using Naïve Bayes and K-nearest neighbor algorithms. The key factors contributing to improving classification performance are proper feature weights and less number of dimensions. To improve feature weights, a novel feature weight modification technique is proposed which is based on sentiment scores of the Synset words of input set of words. And to reduce the number of dimensions, we used Latent Semantic Analysis (LSA) technique. From the results, it is proved that the proposed method of modifying weights gives significant improvement in classification performance.

## 1 Introduction

The Internet has plenty of data with respect to sentiment information. From a user's point of view, people are able to post their opinion about anything through various social media, such as Twitter, Facebook etc., These opinions can be gone through to understand the likes and dislikes of the people towards a particular topic or product. These product reviews are textual information through which the opinion of reviewers for the product can be predicted. This prediction can be used for improving the quality of the product and also it can be used to find a set of people who are interested in buying those products.

Sentiment analysis or Opinion mining of product reviews understands the sentiment of users for a particular product. This sentiment can be positive or negative based on the quality and features of the product and each review can be assigned with one of these two classes based on the opinions expressed through the words present in it. This assignment task can be done with the help of classification algorithms.

Performance of review classification can be improved by computing proper weights for the features and by reducing the number of features. As the feature weights do not include synonyms and sentiment values, it is tough to have accurate classification. Also, there are more number of features present in the reviews, classification performance may get degraded. This

has motivated us to introduce a novel technique for improving feature weights and to use dimensionality reduction techniques to reduce the number of dimensions.

The key contributions of the proposed work are three fold:

- i. It proposes a novel feature weight modification based on Synset and sentiment scores of the features.
- ii. It uses Latent Semantic Analysis (LSA) technique to reduce the number of dimensions.
- iii. The proposed feature weight model using LSA is compared with traditional feature weight model and analyzed through experiments.

The remainder of the paper is organized as follows: Section 2 discusses various techniques in the area of product review classification. Section 3 states the problem. Section 4 explains various stages in the proposed method. Section 5 presents the results obtained. Section 6 concludes the paper.

## 2 Literature review

Product review classification finds the feelings of reviewers by assigning class for each review as positive or negative. The extracted information tells the quality of the product and it can be used to improve the product. This analysis can also be used to automate the task of assigning the polarity for the future reviews and suggesting the same kind of products to the users. Many have done their research in the area of sentiment analysis of reviews.

Rai and Mewada, 2017 [1] have proposed a technique to select efficient features based on the polarity of the movie reviews i.e., positive or negative. Fuzzy clustering-based classifiers are a suitable solution for detecting polarity due to their capability to capture the qualitative and semantic elements of similarity. Among different classification techniques, it is found that the highest accuracy was achieved by Random Forest with an accuracy of 99.89%.

Dey et al., 2020 [2] have made a comparative study of Support Vector Machine (SVM) and Naive Bayes Classifier for Sentiment Analysis on Amazon Product Reviews. The model proposed by them generates 84% and 82.875% accuracy on SVM and Naive Bayes accordingly which is confirmed to be leading compared to the conventional techniques. Experimental results have confirmed that the SVM can polarize the feedback of Amazon products with a higher accuracy rate.

Mamtesh and Mehla 2019 [3] have proposed a technique in which data sets are extracted from heterogeneous sources and derived relevant tokens by applying train test split and count vectorization techniques. Three Machine Learning algorithms namely, K Nearest Neighbours, Logistic Regression, Naive Bayes are implemented in order to train the data set, to analyse the reviews and predicted the sentiment of the reviews either positive or negative with high accuracy and they concluded that Naive Bayes can be used efficiently and appropriately in implementing sentiment analysis.

Multilabel classification is proposed by Liu and Chen, 2015 [4] for sentiment classification. Classification is done by first dividing the problem into multiple single label problems. The set of new problems are considered for training the model. Then the trained model is used for predicting the class of the text using single labels and it is translated into multiple labels.

Tripathy et al., (2015) [5] have made an attempt to classify sentiment reviews using supervised machine learning algorithms. Four supervised algorithms such as Naive Bayes (NB), Support Vector machine (SVM), Random Forest and Linear Discriminant Analysis

(LDA) have been considered in their work. These algorithms were implemented on two different datasets (IMDb and polarity), as being considered by different authors. It was observed that out of four algorithms, the algorithm, using Random Forest classifier yields more convincing results in comparison with other classifiers on both datasets.

Tsutsumi et al., 2007 [6] have compared the performance of SVM, Maximum Entropy and Scoring methods when classifying movie reviews. 1400 reviews were considered in this work in which 700 are positive reviews and 700 are negative reviews. It is proved in the paper that SVM performs better than other two techniques.

Rios et al., 2015 [7] have compared all the latent semantic models namely Latent Semantic Analysis (LSA), Latent Dirichlet Allocation (LDA), Pitman - Yor Topic Model (PYTM), Non-Negative Matrix Factorization (NMF) for the applications of industry.

Ramanathan et al., 2019 [8] have developed a model to determine the sentimental opinion and category of Tamil movies based on twitter messages. In their research work, TFIDF method is used to find the accuracy based on keywords. To improve the performance, they have applied domain specific ontology. They have used Tamil SentiWordNet with adjectives to classify the sentiment.

Peter et al., 2009 [9] have made a study on the performance of dimensionality reduction methods such as SVD and NMF on the MEDLINE dataset. Both the methods have well performed over the conventional Vector Space Model. NMF has the advantage over SVD that it produces a natural “additive parts-based” representation of data, owing to its non-negativity, which can be helpful in document clustering based on topics.

Fernandez et al., 2016 [10] proposed a method for analyzing sentiments in texts based using unsupervised dependency parsing-based classification technique that utilizes a combination of NLP techniques and features which are derived from lexicons. These sentiment lexicons were constructed by means of a semiautomatic polarity expansion algorithm in order to improve results in specific application domains.

Liu et al., 2013 [11] used SVM as a classifier for classifying sentiments. They suggested that subject of the opinion and the credibility of the person who gives opinion are also much important for classification. Orkphol and Yang, 2019 [12] performed clustering of microblogs using K-means algorithm. They selected the initial set of centroids using artificial bee colony algorithm. Poomagal et al., 2020 [13] proposed a technique for selecting centroids for K-means based on sentiment scores. They have also grouped the features based on their similarity.

### **3 Problem statement**

Given a set of reviews, the proposed method classifies these reviews into positive or negative by first modifying the feature values using a novel feature weight improvement model which is based on Synset and sentiment scores of the features in the collection. Also, the number of dimensions in the model is reduced using LSA

### **4 Proposed Method**

Proposed method classifies the product reviews by first modifying the weight of the words using their sentiment scores and applying dimensionality reduction techniques. LSA is used for reducing the dimensions and classification is done using Naïve Bayes and K-nearest neighbor algorithms. The steps in the proposed method are as follows,

- i. Preprocessing
- ii. TFIDF weight calculation
- iii. Weight modification using sentiment scores
- iv. Latent Semantic Analysis (LSA)
- v. Classification algorithms

#### 4.1 Preprocessing

Preprocessing of reviews involves the removal of punctuations, stop words, and collecting useful features. Let  $R = \{r_i\}_{i=1}^n$  be the set of reviews and  $W = \{w_i\}_{i=1}^m$  be the set of features collected from the reviews.

#### 4.2 TFIDF Weight calculation

Once the features are extracted from the reviews, vector space model is to be constructed for further processing. One of the basic vector space model is bag of words model in which the entries may be 1 or 0 based on the occurrence of the feature inside the review. Another way is to enter the frequency (Term Frequency - TF) of the features in the model. This model has a flaw that very common words may have high frequency. But they do less contribution for differentiating reviews. To overcome this flaw, another weight (Inverse Document Frequency - IDF) can be calculated and included with TF which increases the weight of less common terms and reduces the weight of more common terms. When a feature occurs in all the reviews, it will have 0 as its IDF value. TF is calculated using Equation (1).

$$TF = \frac{\text{Number of times the word occurs in the document}}{\text{Total number of words in the document}} \quad (1)$$

Inverse Document Frequency measures how important a term is. Frequent terms are weighed down and rare terms are weighed up. IDF is calculated using Equation (2).

$$IDF = \frac{\text{Total number of documents}}{\text{Number of documents with the word in it}} \quad (2)$$

TF of a word multiplied with its IDF gives the TFIDF value of the word, which is used to measure the relevance of the word in the document. TFIDF is calculated using Equation (3).

$$TFIDF = TF * IDF \quad (3)$$

#### 4.3 Weight modification using sentiment scores

The proposed novel method of modifying TFIDF values is based on Synset and Sentiment scores of the features. This task is done by first finding Synset of the words in each review and then extracting sentiment scores of those Synset of the words and modifying the TFIDF value based on the positive sentiment score value or negative sentiment score value. Here the sentiment polarity for each entry in the TFIDF matrix is assigned based on the polarity of the features present in it and occurrence of it in the respective review. For finding sentiment scores, SentiWordNet is used which returns three different scores for each feature namely positive, negative and neutral. The summation of three scores is equal to 1. If the summation of positive scores for synset of the word is greater than summation of negative scores then the respective summation of positive score is averaged and the average score is added in the respective entry in TFIDF matrix. Otherwise, the averaged negative score is added in the respective entry in TFIDF matrix. This process is mathematically explained using Equations (4) to (14).

Let  $PS_{ij}$  represents the summation of positive sentiment scores of features of  $i^{\text{th}}$  review in  $j^{\text{th}}$  word whose positive sentiment score is higher than negative sentiment score,  $Pavg_{ij}$  be the average positive sentiment scores and  $PC_{ij}$  be the count of the synset having more positive score than negative score. Let  $NS_{ij}$  represents the summation of negative sentiment scores of features of  $i^{\text{th}}$  review in  $j^{\text{th}}$  word whose negative sentiment score is higher than positive sentiment score,  $Navg_{ij}$  be the average negative sentiment scores and  $NC_{ij}$  be the count of the synset having more negative score than positive score.  $TFIDF_{ij}$  represents the TFIDF score of  $j^{\text{th}}$  word in  $i^{\text{th}}$  review.

$$(4) \quad PS_{ij} = \begin{cases} PS_{ij} + \text{sentiposscore}(w_k), & \text{if } w_k \in \text{synsets}(j), TFIDF_{ij} \neq 0, \\ & \text{sentiposscore}(w_k) > \text{sentinegscore}(w_k) \\ & \text{and } 1 \leq k \leq \text{len}(\text{synsets}(j)) \\ 0 & , \text{if } TFIDF_{ij} = 0 \end{cases}$$

$$(5) \quad NS_{ij} = \begin{cases} NS_{ij} + \text{sentinegscore}(w_k), & \text{if } w_k \in \text{synsets}(j), TFIDF_{ij} \neq 0, \\ & \text{sentiposscore}(w_k) < \text{sentinegscore}(w_k) \\ & \text{and } 1 \leq k \leq \text{len}(\text{synsets}(j)) \\ 0 & , \text{if } TFIDF_{ij} = 0 \end{cases}$$

where  $1 \leq i \leq n$  and  $1 \leq j \leq m$

$$\text{syn} = \text{len}(\text{synsets}(j)) \quad (6)$$

$$\forall 1 \leq i \leq n, 1 \leq j \leq m PC_{ij} = NC_{ij} = 0 \quad (7)$$

$$(8) \quad PC_{ij} = \sum_{k=1}^{\text{syn}} 1 \left( w_k \in \text{synsets}(j), TFIDF_{ij} \neq 0, \text{sentiposscore}(w_k) > \text{sentinegscore}(w_k) \right)$$

$$(9) \quad NC_{ij} = \sum_{k=1}^{\text{syn}} 1 \left( w_k \in \text{synsets}(j), TFIDF_{ij} \neq 0, \text{sentiposscore}(w_k) < \text{sentinegscore}(w_k) \right)$$

$$(10) \quad \text{poscount} = \sum_{k=1}^{\text{syn}} 1(\text{sentiposscore}(w_k) = 0)$$

$$(11) \quad \text{negcount} = \sum_{k=1}^{\text{syn}} 1(\text{sentinegscore}(w_k) = 0)$$

where  $1(f)$  returns 1 if  $f$  is true and  $1 \leq i \leq n$  and  $1 \leq j \leq m$

$$(12) \quad Pavg_{ij} = \frac{PS_{ij}}{\text{syn} - \text{poscount}}$$

$$(13) \quad Navg_{ij} = \frac{NS_{ij}}{\text{syn} - \text{negcount}}$$

$$TFIDF_{ij} = \begin{cases} TFIDF_{ij} + Pavg_{ij}, & \text{if } PC_{ij} > NC_{ij} \\ TFIDF_{ij} + Navg_{ij}, & \text{Otherwise} \end{cases}$$

(14)

**Pseudo Code :**

```

fori=1 to n do
  for j=1 to m do
    val = TFIDF[i][j]
    ifval≠ 0 then
      word = u_words[j]
      ls = synsets(word)
      pos = 0, neg = 0
      iflen(ls) ≠ 0 then
        ps = 0, ns = 0, pc = 0, nc = 0, poscount = 0, negcount = 0
        forsyn in ls do
          ifsentiposscore[syn] = 0 then
            poscount = poscount + 1
          ifsentinegscore[syn] = 0 then
            negcount = negcount + 1
          ifsentiposscore[syn] > sentinegscore[syn] then
            ps = ps + sentiposscore[syn]
            pc = pc + 1
          else
            ns = ns + sentinegscore[syn]
            nc = nc + 1
          pos = ps / (len(ls) - poscount)
          neg = ns / (len(ls) - negcount)

        if pc > nc then
          val = val + pos
        else
          val = val + neg
        TFIDF[i][j] = val

```

#### 4.4 Latent Semantic Analysis (LSA)

Once the TFIDF values are altered, feature set reduction of reviews is done using LSA. LSA learns latent topics by performing a matrix decomposition on the document-term matrix using SVD. With SVD, the original matrix is decomposed into three matrices namely  $U_{n \times k}$ ,  $\Sigma_{k \times k}$ ,  $V^*_{k \times m}$ . Here we are using truncated SVD for reducing the dimension where the value of k is specified and then altered TFIDF matrix is decomposed. From those three matrices,  $U_{n \times k}$  is taken as a resultant dimension reduced matrix. LSA uses a bag of word (BoW) model, which results in a term-document matrix (occurrence of terms in a document). Rows represent reviews and columns represent terms. LSA learns latent topics by performing a matrix decomposition on the document-term matrix using SVD (Singular Value Decomposition). LSA is typically used as a dimension reduction or noise reducing technique. SVD is a matrix factorization method that represents a matrix as the product of three matrices. It is represented in Equation (15).

$$M = U\Sigma V^* \quad (15)$$

Where M is a  $n \times n$  matrix, U is a  $n \times m$  left singular matrix,  $\Sigma$  is a  $m \times m$  diagonal matrix with non-negative real numbers, V is a  $n \times m$  right singular matrix,  $V^*$  is a  $m \times n$  matrix, which is the transpose of V.

## 4.5 Classification Algorithms

Once the dimensionality reduction is performed, classification of reviews is done using two classification algorithms namely, NB and k-NN. NB algorithm calculates the probability of each review belonging to positive class or negative class. Then it assigns the class of higher probability to the review. In this paper, Gaussian NB is used to find the class of the review as the feature values are continuous. In k-NN algorithm, k nearest reviews for each test review are considered and the majority of the class of those reviews is assigned.

### 4.5.1 NAIVE BAYES

Naive Bayes (NB) classifiers are a collection of classification algorithms based on Bayes' Theorem with a common assumption of conditional independence between every pair of features. The different NB classifiers differ mainly by the assumptions they make regarding the distribution of  $P(x_i / y)$ .

Gaussian NB is a NB classifier dealing with continuous data, with an assumption that the continuous values associated with each class are distributed according to a normal or Gaussian distribution. The likelihood of the features is assumed to be Gaussian as given in Equation (16).

$$P(X|y) = \frac{1}{2\pi^2|\Sigma|} \exp(-(X - \mu)\Sigma^{-1}(X - \mu)^T) \quad (16)$$

Where X is the data point, y is class label,  $\mu$  is the mean of the data points associated with class y and  $|\Sigma|$  is the determinant of covariance matrix of class k and  $\Sigma^{-1}$  is the inverse of covariance matrix.

### 4.5.2 k – NEAREST NEIGHBOUR Algorithm

The k-Nearest Neighbour algorithm (k-NN) is a non-parametric method used for classification and regression. The k-NN classification algorithm classifies the input data based on its similarities with the training data. In this algorithm, distance between test data and each row of training data is calculated with the help of any distance measures such as Euclidean, Manhattan or Hamming distance. Then the distance value along with the data, is sorted in ascending order. The test data is assigned a class based on the most frequent class of the top k rows of the sorted array. The value of k can be any positive integer, usually a small value.

## 5 Experimental results

In order to measure the performance of the proposed model, experiments are being conducted using ten labelled datasets which are taken from the Multi-Domain Sentiment Dataset Repository (<https://www.cs.jhu.edu/~mdredze/datasets/sentiment/>) and the results are

reported. Each dataset contains 1000 positive and 1000 negative reviews. All the datasets used in this work are sentiment based since they project positivity and negativity of the products.

### 5.1 Experimental setup

Experiments are performed in the following 4 scenarios:

M1 Unmodified TFIDF without Reduction

M2 Modified TFIDF using Sentiment Scores without Reduction

M3 Unmodified TFIDF with LSA Reduction Technique

M4 Modified TFIDF using Sentiment Scores with LSA Reduction Technique

In each scenario, experiments are performed and the results are presented. All these comparisons are performed using the metrics namely accuracy and F-measure. Mathematical equations of these metrics are presented in Equations (17) and (18).

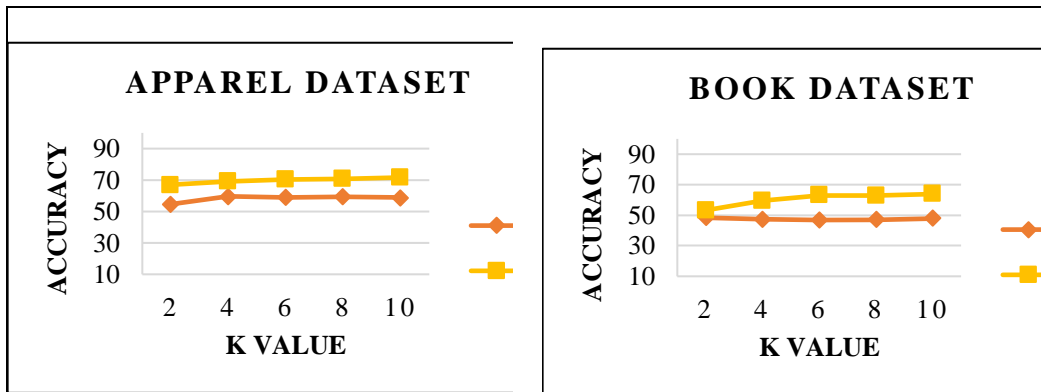
$$Accuracy = \frac{TP+TN}{Total} * 100 \tag{17}$$

$$F - measure = \frac{2*Precision*Recall}{Precision+Recall} \tag{18}$$

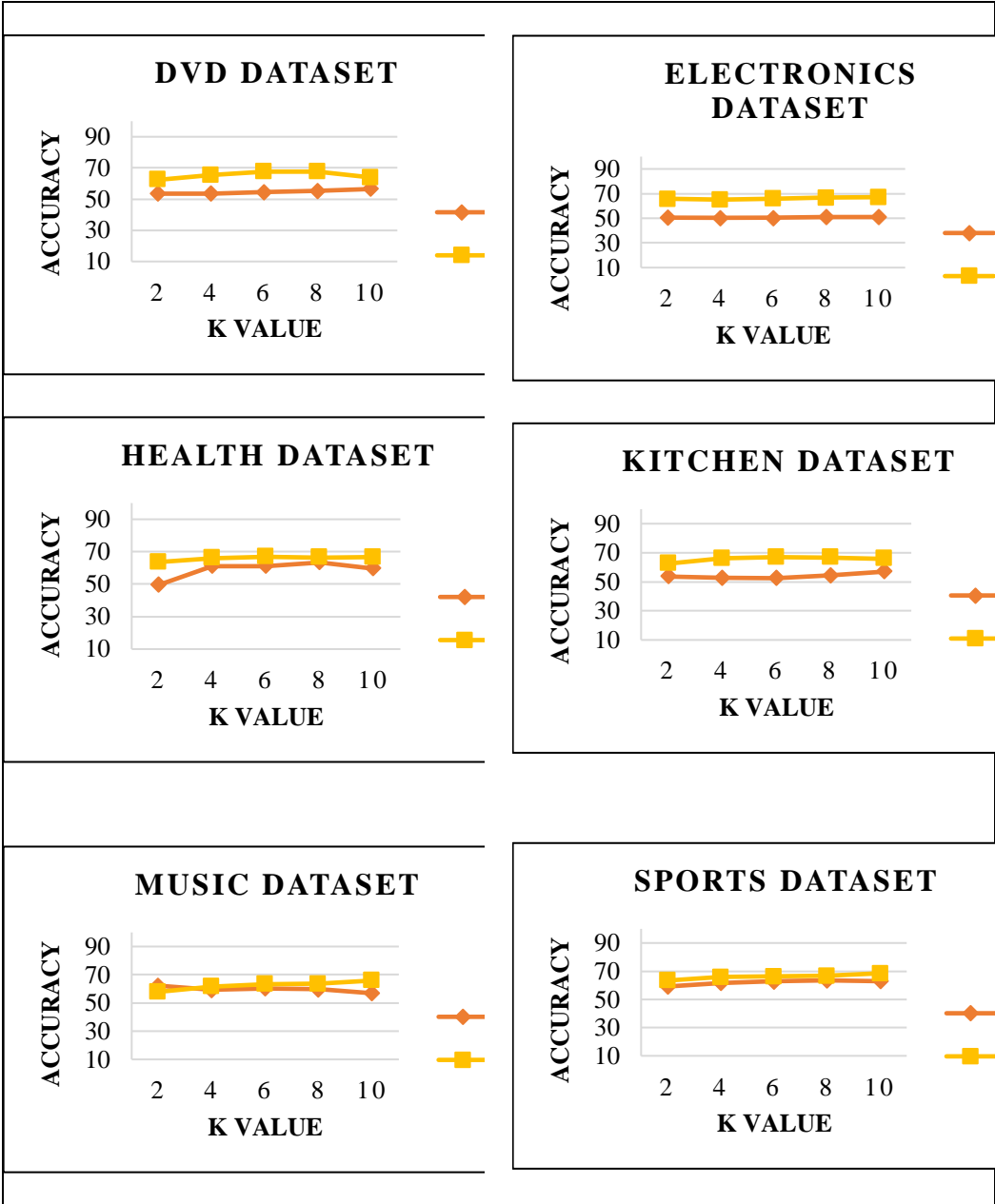
Where TP, TN and Total represent True Positive, True Negative and total number of reviews in the collection respectively.

### 5.2 Comparative study

For comparing the results of proposed feature weight improvement technique, two cases namely No Reduction and NMF Reduction were applied on 10 datasets of product reviews. In each case, we have two methods namely unmodified TFIDF and modified TFIDF value using the sentiment scores. NB and k-NN algorithms were executed in both the cases and the results were noted. k-NN algorithm is executed by varying k value and the accuracy obtained with 5 different k values are presented in Figure 1.







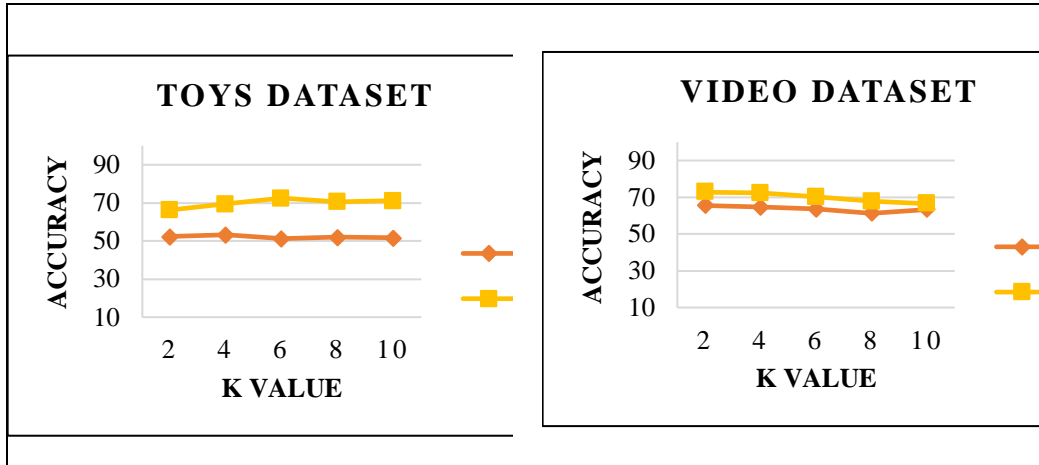


Figure 1 Comparison of M1 and M2 (No Reduction) by executing k-NN

It is observed from Figure 1 that the proposed method M2 gave better results than M1 for 9 datasets for all k values. In the first case, after modifying TFIDF vectors, accuracy is increased for all datasets using k-NN classifier. For the remaining 1 dataset, few values of k have produced better results in case of M1. Overall, M2 outperforms M1 for all the cases. The reason for this increase in accuracy produced by M2 is due to the updation of feature weights by using sentiment scores of words in reviews. Reviews with more positive/negative sentiment scores will get their respective weight value increased.

The performance of proposed feature weight improvement method M2 is compared with conventional method M1 using Naïve Bayes and k-NN in terms of accuracy and F-measure and the results are projected in Figures 2 and 3.

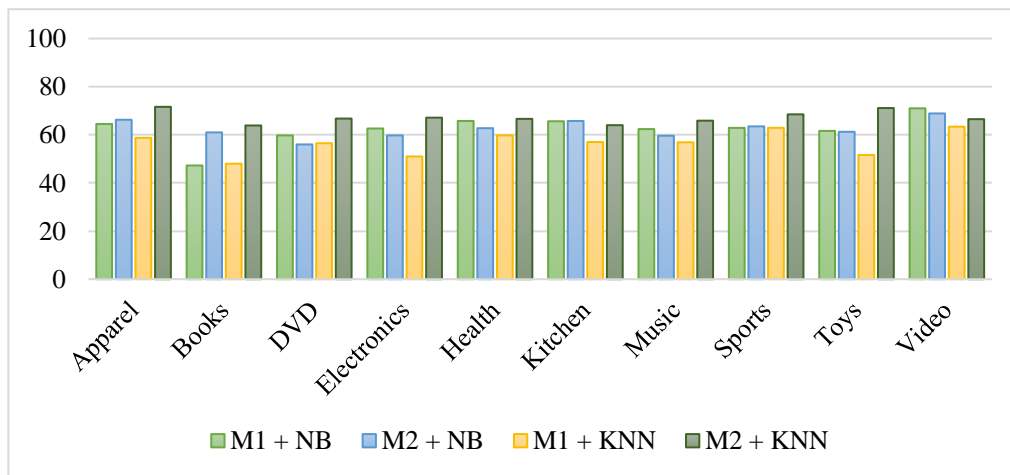


Figure 2 Comparison of M1 and M2 (No Reduction) using Accuracy

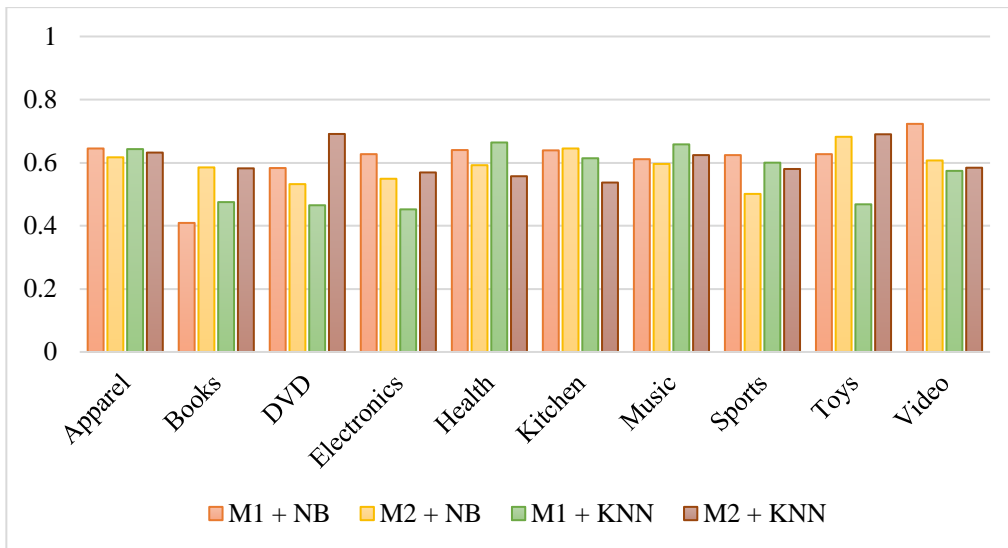
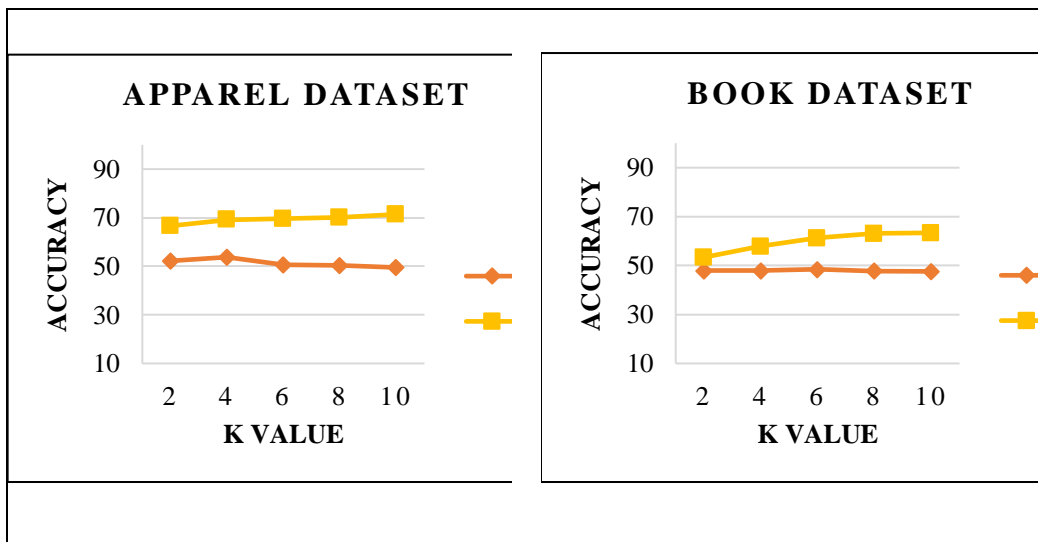
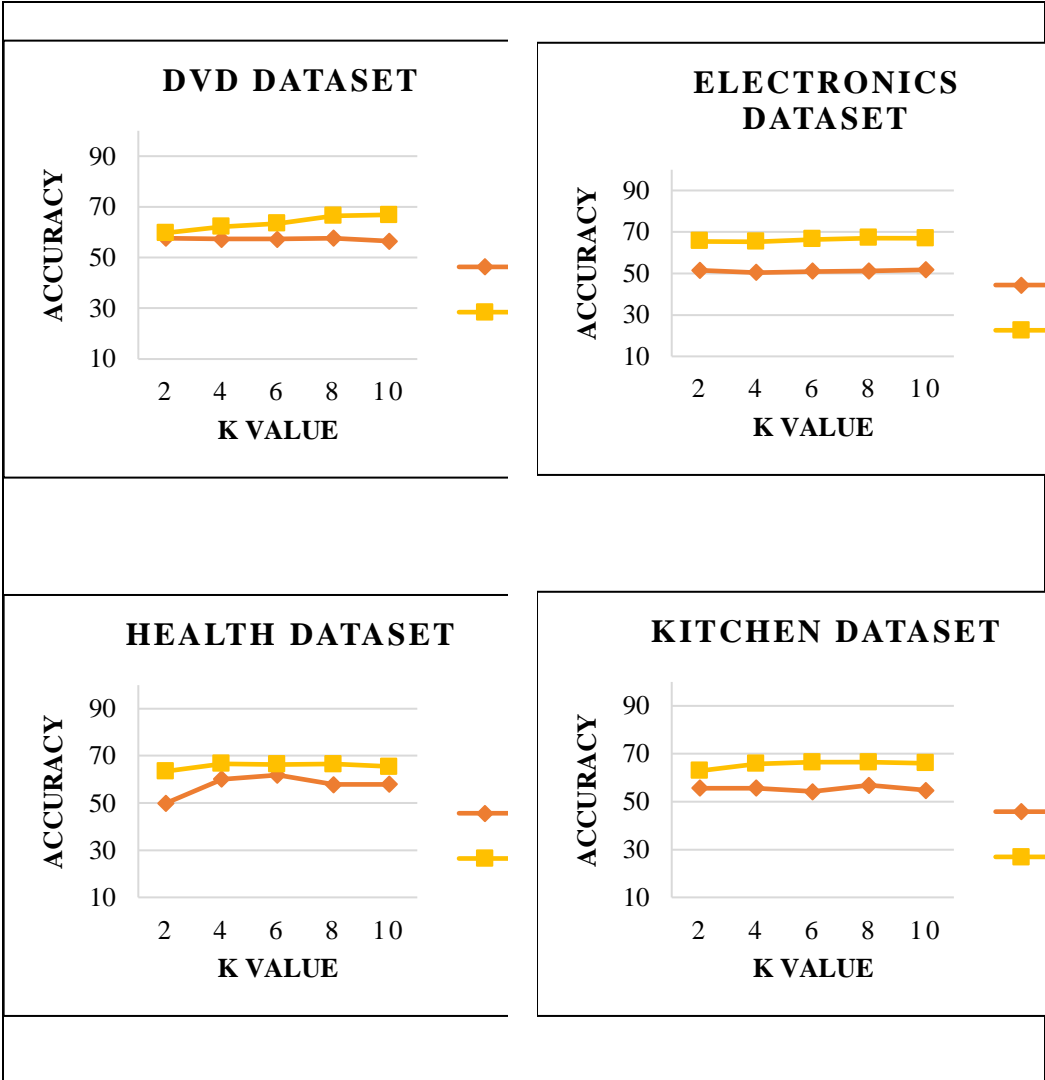


Figure 3 Comparison of M1 and M2 (No Reduction) using F – Measure

Similarly for the second case, dimensionality reduction using LSA technique is carried out and it is observed that accuracy is increased for 9 out of 10 datasets using k-NN classifier. The results of M3 and M4 with k-NN classifier are shown in Figure 4.





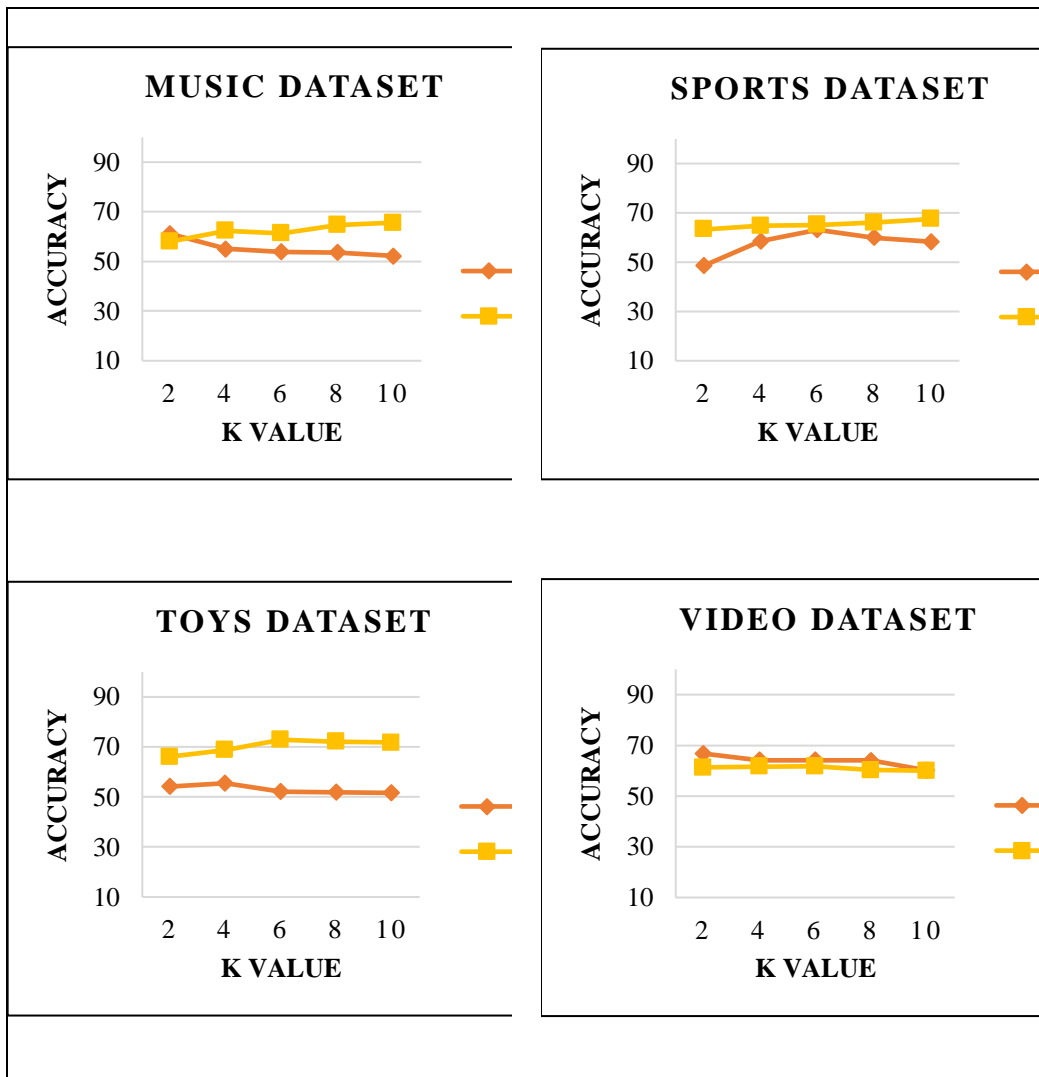


Figure 4 Comparison of M3 and M4 (LSA Reduction) by executing k-NN

Comparison of performance between proposed feature weight improvement method M4 (after applying LSA) with the conventional method M3 (after applying LSA). Classification is performed and the results are projected in Figure 5 and 6.

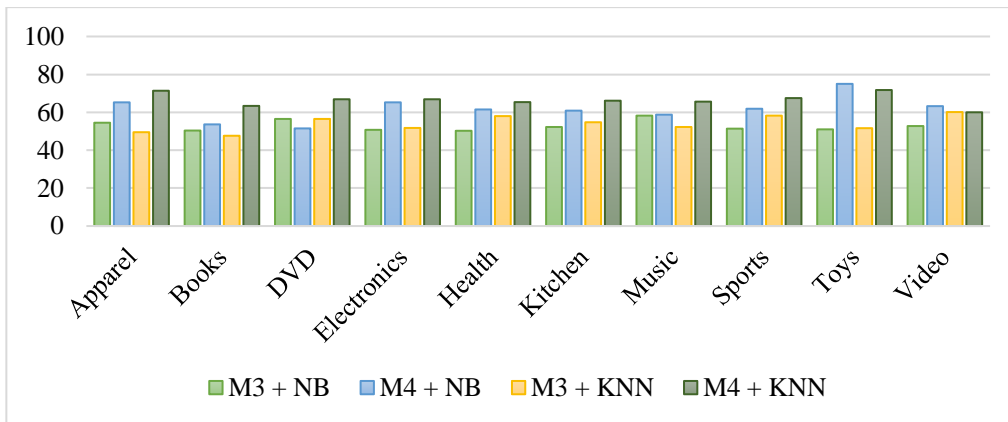


Figure 5 Comparison of M3 and M4 (LSA Reduction) using Accuracy

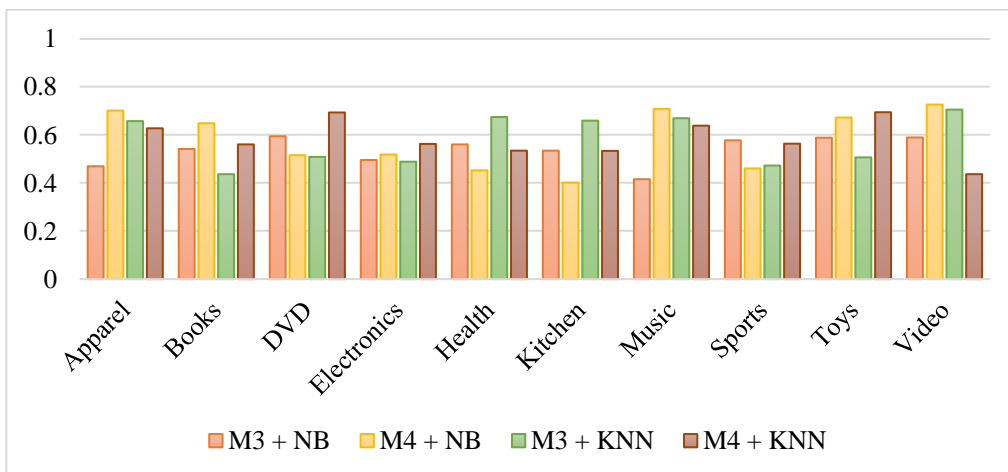


Figure 6 Comparison of M3 and M4 (LSA Reduction) using F - Measure

The summary of the results obtained before applying LSA and after applying LSA is presented in Table 1. It specifies the method which produces better results for each dataset in each case.

Table 1 Summary of results before and after applying LSA

Dataset	Before Reduction		After Reduction	
	Accuracy	F-Measure	Accuracy	F-Measure
Apparel	M2+k-NN	M1+NB	M4+k-NN	M4+NB

Books	M2+k-NN	M2+NB	M4+k-NN	M4+NB
DVD	M2+k-NN	M2+k-NN	M4+k-NN	M4+k-NN
Electronics	M2+k-NN	M1+NB	M4+k-NN	M4+k-NN
Health	M1+NB	M1+k-NN	M4+k-NN	M3+k-NN
Kitchen	M2+NB	M2+NB	M4+k-NN	M3+k-NN
Music	M2+k-NN	M1+k-NN	M4+k-NN	M4+NB
Sports	M2+k-NN	M1+NB	M4+k-NN	M3+NB
Toys	M2+k-NN	M2+NB	M4+k-NN	M4+k-NN
Videos	M1+NB	M1+NB	M4+NB	M4+NB

It is observed from Table 1 that the proposed feature weight improvement model performs well in 80% of the cases and 40% of the cases in terms of accuracy and F-measure respectively before reducing dimensions. The performance is further improved after applying LSA and it is shown that accuracy is high in all the datasets and F-measure is high in 70% of the dataset. From the analysis, it can be concluded that the improvement in the results is achieved due to the modification of feature weights.

## Conclusion

Improvement in sentiment analysis of product reviews is done using a technique which is based on the modification of feature weight vector using the sentiment scores of Synset words. Also, dimensionality reduction is done using LSA technique. From experimental results, it is shown that the proposed method outperformed the conventional method in most of the cases. The proposed method achieved the maximum accuracy of about 72% before and after applying LSA technique. From all the observations, it is concluded that the improvement of feature weight makes a huge impact on classification results. In future, this work can be extended by considering the synonym similarity of words and also bigram representation can be considered for better results.

## References

- [1] Rai R and Mewada P, (2017), "Sentiment Analysis of Movie Review using Machine Learning Approach", International Journal Online of Sports Technology & Human Engineering, Vol.4, No. 4, pp. .
- [2] Dey S, Wasif S and Sultana S, "A Comparative Study of Support Vector Machine and Naïve Bayes Classifier for Sentiment Analysis on Amazon Product Reviews", Proceedings of International Conference on Contemporary Computing and Applications (IC3A), 2020., 2020.
- [3] Mamteshand Mehla S, (2019), "Sentiment Analysis of Movie Reviews using machine Learning Classifiers", International Journal of Computer Applications, Vol. 182, No. 50.

- [4] Liu S M and Chen J H,(2015) ,“A multi-label classification-based approach for sentiment classification”, Expert Systems with Applications, Vol. 42, No.3, pp. 1083-1093.
- [5] Tripathy A,Agarwal A and Rath S K, (2017), “Classification of Sentiment of Reviews using Supervised Machine Learning Techniques”,Procedia Computer Science, Vol. 57, pp. 821-829.
- [6] Tsutsumi K, Shimada K and Endo T,“Movie Review Classification on a Multiple Classifier”,Proceedings of the 21st Pacific Asia Conference on Language, Information and Computation,2007, pp. 481-488.
- [7] Rios S A,(2016), “An Empirical Comparison of Latent Semantic Models for Applications in Industry”,Neurocomputing, Vol. 179, pp.176-185.
- [8] Ramanathan V, Meyyappan T andThamarai S M, (2019), “Predicting Tamil Movies Sentimental Reviews using Tamil Tweets”, Journal of Computer Science, Vol. 15, No. 11, pp. 1638-1647.
- [9] Peter R, Shivapratap G, Divya G and Soman K P,(2009), “Evaluation of SVD and NMF Methods for Latent Semantic Analysis”,International Journal of Recent Trends in Engineering, Vol. 1, No. 8, pp. 808-810.
- [10] Fernandez G M, Alvarez L T, Juncal M J, Costa M E and Gonzalez C F J, (2016), “Unsupervised method for sentiment analysis in online texts”, Expert Systems with Applications, Vol. 58, pp. 57-75.
- [11] Liu H, He J, Wang T, Song W and Du X, (2013),“Combining user preferences and user opinions for accurate recommendation”, Electronic Commerce Research and Applications. Vol. 12, No. 1, pp. 14–23.
- [12] Orkphol K and Yang W,(2019), “Sentiment Analysis on Microblogging with K-means Clustering and Artificial bee colony”, International Journal of Computational Intelligence and Applications. Vol. 18, No. 03, pp. 1-22.
- [13] Poomagal S, Malar B, Inamul Hassan and Kishor R, (2020), “A novel Tag\_Score (T\_S) model with improved K-means for clustering tweets”, Sadhana : Indian Academy of Sciences, Article ID : 0125, pp. 1-13.