Data Classification Using Machine Learning Approach

Shekhar Pandey(✉), Supriya M, and Abhilash Shrivastava

Department of Computer Science and Engineering,
Amrita School of Engineering, Amrita Vishwa Vidyapeetham,
Amrita University, Bengaluru, India
chirag1989shekhar@gmail.com,
m_supriya@blr.amrita.edu,
shrivastava.abhilash2510@gmail.com

Abstract. Currently, Internet has numerous effects on our everyday lifecycle. Its significance as an intermediate for commercial transactions will develop exponentially throughout the next years. In terms of the engaged marketplace volume, the Business to Business region will hereby be the supreme exciting area. As the extensive usage of electronic business transactions increase, great volume of products information gets generated and managing such large information automatically becomes a challenging task. The accurate classification of such products to each of the existing classes also becomes an additional multifarious task. The catalog classification is an essential part for operative electronic business applications and classical machine learning problems. This paper presents a supervised Multinomial Naïve Bayes Classifier machine learning algorithm to classify product listings to anonymous marketplaces. If the existing products are classified under the master taxonomy, the task is to automatically categorize a new product into one of the existing categories. Our algorithm approach proposes a method to accurately classify the existing millions of products

Keywords: Naïve Bayes · Classifier · Machine learning · Categories

1 Introduction

Small scale to giant scale sized businesses who trade products online spend a substantial part of their time, money, and struggle - in categorizing the products they trade, to better market their products, and in determining which products to sell. Such E-inventory (Electronic index) businesses hold their data of items and administrations in a web based business association. E-list is a type of classification in which information of an inventory is categorized to one of the already existing classes list. The classes are categorized by a definite taxonomy framework which as a rule has an arranged structure. Accurate taxonomy is essential not just for information introduction and synchronization among business accomplices, additionally to keep the quickly expanding item information viable and adequate. Still, merchandise data cataloguing is an extremely tedious job and not easy to do by hand due to its enlarged product information. In this

S.M. Thampi et al. (eds.), *Intelligent Systems Technologies and Applications*,
Advances in Intelligent Systems and Computing 683,
https://doi.org/10.1007/978-3-319-68385-0_10

paper, the use of automatic learning techniques has been proposed to outline product classes (e.g., ‘Electronics’) and potential subcategories (e.g., ‘Printers’). This is beneficial for the circumstance where a business has a list of new products that has to be sold by automatically classifying based on training data of the businesses’, other products and classifications. This will also be beneficial for categorizing a new merchandise item line that has not been previously introduced in the market before, or for the items that are more densely populated than the training data set. To advance this procedure, an automatic learning algorithm has been proposed that can automatically categorize listings with high accuracy. A number of competitor customary classification schemes are already available in the market but none of them are globally recognized and accepted [1]. Works in [2] connected a few procedures from data recovery and machine learning approaches, figuring out how to proceed with item information characterization by incorporating striking calculations like KNN (K-Nearest Neighbor), SVM (Support Vector Machine) and NBC (Naïve Bayes classifier) etc.

2 Related Works

Currently, with the fast development of small-organized documents, their taxonomy categorization issue has pulled in an expanding consideration. The normal motivation is that the arrangement of such large documents may comprehend helpful data for classification. There is a need for several attempts to analyze anonymous marketplaces [3]. In [3], the author analyses on the Silk Road for 8 months, investigating product entries and the complete distribution of product listings. Be that as it may, they depended on seller provided classifications, which does not exist for all commercial centers. Also, a few sellers purposefully misclassify their item to seem higher in commercial center query output. Correcting for these problems, the categorization of taxonomy [4] has been constructed on the Bayesian networks. For each document in the preparation set, it amasses a Bayesian system whose structure is basically the same as that of the record itself appeared as a tree. Constructed on Bayesian networks conditional probability, the work proposed in [4] develops the classification of an information archive. Zaki et al. [5] describe a technique for construction for XML taxonomy classifier based on administrator and Denoyer et al. [6] recommend a classifier which classifies structured multimedia taxonomy based on Bayesian. Vinithra et al. [7], Ani et al. [8] and Priyanka et al. [9] also outlines the various classification techniques. Most of the automatic learning methods are well documented in the literature as effective binding blocks for document classification systems.

Motivated by the study from different researchers, this work decides to deal with Multinomial Naïve Bayes Classifier to making a document classifier.

Naïve Bayes approach works on each word position which is described to be an attribute of the Naïve Bayes Classifier [10, 11] for level content classification. Moreover, seeing that attribute has distinctive frequency power individually, we indulgence singular attributes diversely by allotting weights as per their significance. Each attributes are normalized before allotting the weights to the attributes sensibly. The frequency method has been used for exactness classification. Our classifier demonstrates enhanced exactness with the Multinomial Naïve Bayes Classifier even if there is noisy data.

3 Classification Algorithm Multinomial Naïve Bayes Classifier

The Multinomial Naïve Bayes is adequately automatic calculation for content classification because of its quality and has good execution performance. We demonstrate this methodology to classify the catalogs rendering to their classes.

3.1 Multinomial Naïve Bayes Classifier

To categorize an item or word w, the Naïve Bayes calculates the posterior probability $P(w|d)$ of that word or item constructed on the Bayes Theorem. Specified a group of classes Z, the attributes $<k_1, k_2,\ldots, k_n>$ and the values $<f_1, f_2,\ldots, f_n>$ that designate an input instance, the Naïve Bayes allocates the most likely classification as specified by the supplementary calculation method.

$$Z_{NB} = \arg max_{w_j \in Z} P(w_j) \prod_i P(k_i = f_i | w_j) \tag{1}$$

where Z_{NB} is Naïve Bayes class.

This method classifies any catalogs which have huge number of attributes based on the word's probability and frequency. This model does not neglect any words even if they have less probability because this model treats all words equally to get accurate results. When this method is used for catalog classification, every written text treats as attribute for classification. Specified each word m, treat as individually $<m_1, m_2,\ldots, m_n>$ that constitutes an input document, the Naïve Bayes can be represented as

$$Z_{NB} = \arg max_{w_j \in Z} P(w_j) \prod_i P(k_i = m_i | w_j) \tag{2}$$

Treating that each word has equal priority and supposition that the elements are indistinguishably conveyed to reduce the cost, the above approach implies that the probability of experiencing each word is independent of the particular word position [4].

3.2 Extending and Normalizing Attributes

Since the qualities of writings are made out of many words and are regularly boisterous, tolerating just the correct matches is deluding. It is plainly wrong to recognize "Laptop" and 'Laptop Notebook'. The issue ends up being more lamentable when we endeavor to use a property like 'item portrayal' which is now and again made out of full sentences. So, sometimes even each word method is not successful because of the same name with different writing styles. Hence, we reclassify the estimation of a property as

$$f_i = \{n_{i1}, n_{i2}, \ldots .n_{iq}\} \tag{3}$$

where n_{iq} is a value formed from f_i by the parser. Then we can reasonably assume that

$$\begin{aligned} Z_{NB} &= \arg max_{w_j \in Z} P\left(w_j\right) \prod\nolimits_{i,j} P(n_{iq}\ appears\ in\ k_i | w_j) \\ &= \arg max_{z \in Z} \left\{ \left|w_j\right| \prod\nolimits_{iq} \frac{n(w_j, k_i, n_{iq})}{n(w_j, k_i)} \right\} \end{aligned} \tag{4}$$

where $n(w_j, k_i, n_{iq})$ is the existences of n_{iq} in k_i of the phrase which indicates of class w$_j$. Similarly, $n(w_j, k_i)$ is the total frequencies of all in k_i of the catalogs that belong to class w$_j$. $n(k_i)$ is the total number of words in k_i (Figs. 1 and 2).

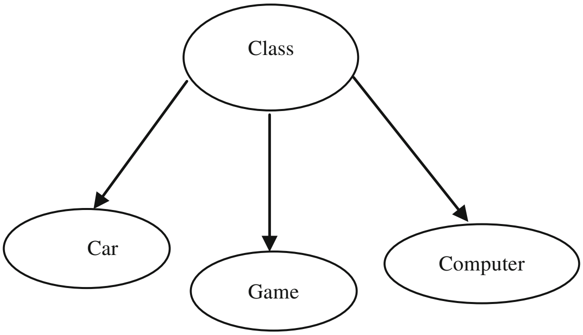


Fig. 1. This showing how we can classify class with unique name from a dataset

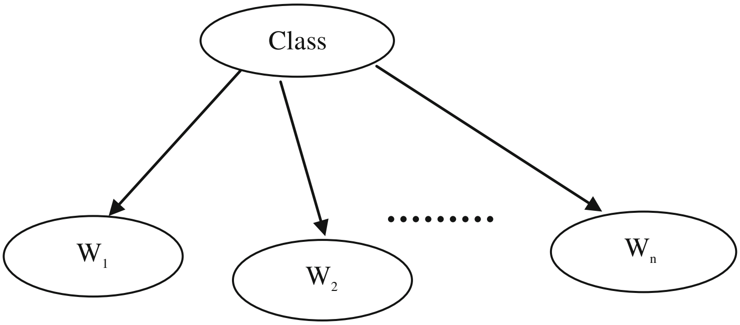


Fig. 2. This show how it will work by taking each word in calculation

The above model works perfectly when text phrases are small and not much bigger, but when some text phrases are big then they will generate more high frequency as compared to small phrases, so we need to overcome it by using the following equation.

$$\begin{aligned} Z_{NB} &= \arg max_{w_j \in Z} P\left(w_j\right) \prod\nolimits_{i} \left(\prod\nolimits_{q} P(n_{iq}\ appears\ in\ k_i | w_j) \right)^{\frac{1}{|f_i|}} \\ &= \arg max_{w \in Z} \left\{ \left|w_j\right| \prod\nolimits_{i} \left(\prod\nolimits_{q} \frac{n(w_j, k_i, n_{iq})}{n(w_j, k_i)} \right)^{\frac{1}{|f_i|}} \right\} \end{aligned} \tag{5}$$

We used the geometric mean for the normalization and after applying the mean the final equation is given as

$$\begin{aligned} Z_{NB} &= \arg max_{w_j \in Z} P(w_j) \prod\nolimits_i \left(\prod\nolimits_q P(n_{iq} \; appears \; in \; k_i | w_j) \right)^{\frac{m_i}{|f_i|}} \\ &= \arg max_{w \in Z} \left\{ |w_j| \prod\nolimits_i \left(\prod\nolimits_q \frac{n(w_j, k_i, n_{iq})}{n(w_j, k_i)} \right)^{\frac{m_i}{|f_i|}} \right\} \end{aligned} \tag{6}$$

where m_i is the weight of the attribute of k_i.

3.3 Applying Multinomial Naïve Bayes Classifier

To change over the test to elements, we used the frequency changes over every token in the listing weight in order to find out how vital that token is to listing; standardized by the quantity of times the token shows up in the entire corpus. This normalization diminishes the effect of basic token in the corpus. To discover any item that has a place with its class, the following steps are to be implemented:

Step 1. Compute the prior probabilities

$$P(category) = \frac{Number \; of \; records \; classified \; into \; the \; category}{Total \; number \; of \; the \; records}$$

Step 2. Compute likelihood.

$$P(word/category) = \frac{Number \; of \; frequency \; of \; a \; word \; in \; all \; records \; from \; a \; category + 1}{All \; the \; words \; in \; every \; document \; from \; a \; category + total \; number \; of \; unique \; words \; in \; all \; the \; records}$$

Step 3. Final computation

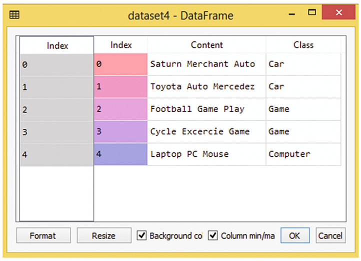
$$\begin{aligned} & P\left({}^{category}/_{records}\right) = \\ & P(category) * P\left({}^{word_1}/_{category}\right) * P\left({}^{word_2}/_{category}\right) * \ldots * \\ & P\left({}^{word_n}/_{category}\right) \end{aligned}$$

The product belongs to the class that has the highest probability among others.

4 Implementation Multinomial Naïve Bayes Classifier

4.1 Training (Step 1 and Step 2 from Sect. 3.2)

While training the dataset of different classes, we count each word as individual and then form the frequency and probability based on number of same occurrence of that word. The dataset Fig. 3 shows that there are more than two classes like Car, Game, and Computer.



Index	Index	Content	Class
0	0	Saturn Merchant Auto	Car
1	1	Toyota Auto Mercedez	Car
2	2	Football Game Play	Game
3	3	Cycle Excercie Game	Game
4	4	Laptop PC Mouse	Computer

Fig. 3. Sample dataset of products which has three classes

4.2 Dataset

For clear description and ease of presence of the method, a test dataset has been shown in Fig. 3. But, the size of the actual dataset is 36256 KB and has been used for validation of the method. The actual dataset can be found in the link (https://github.com/sam-chirag/Data-Classification-Using-Machine-Learning-Dataset)

After applying above Multinomial Naïve Bayes Classifier method (step 1 and step 2) as given in Sect. 3.2, we will get frequency table with their probability as shown in Fig. 5.

4.3 Classification

An example of simple phrase is given below based on above training dataset shown in Fig. 4.

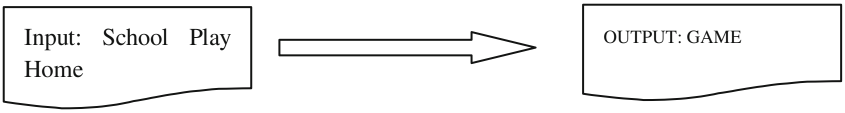


Fig. 4. Input catalog for classification

Classes: Car, Game, Computer

Class Car Probability:

$$P(Car/S1) = P(Car) * P(School/Car) * P(Play/Car) * P(Home/Car)$$

$$P(Car) = \frac{2}{5} = 0.4$$

$$P(School/Car) = \frac{(0+1)}{(6+13)} = 0.05263$$

$P(Play/Car) = 0.05263$ (This word exist in the wordlist of car and we directly taken its likelihood probability value from that table as shown in Fig. 5)

$$P(Home/Car) = \frac{(0+1)}{(6+13)} = 0.05263$$

So,

$$P(Car/S1) = 0.4 * 0.05263 * 0.05263 * 0.05263 = 0.00005825$$

Class Game Probability:

$$\begin{aligned} P(Game/S1) = {} & P(Game) * P(School/Game) * P(Home/Game) \\ & * P(Computer/Game) \end{aligned}$$

$$P(Game) = \frac{2}{5} = 0.4$$

$$P(School/Game) = \frac{(0+1)}{(6+13)} = 0.05623$$

$$P(Play/Game) = 0.10$$

$$P(Home/Game) = \frac{(0+1)}{(6+13)} = 0.05623$$

So,

$$P(Game/S1) = 0.4 * 0.05623 * 0.105 * 0.05623 = 0.00011633$$

Class Computer Probability:

$$\begin{aligned} & P(Computer/S1) \\ & = P(Computer) * P(School/Computer) * P(Play/Computer) \\ & * P(Home/Computer) \end{aligned}$$

$$P(Computer/S1) = \frac{1}{5} = 0.2$$

$$P(School/Computer) = \frac{(0+1)}{(3+13)} = 0.0625$$

$$P(Play/Computer) = 0.0625$$

$$P(Home/Computer) = \frac{(0+1)}{(3+13)} = 0.0625$$

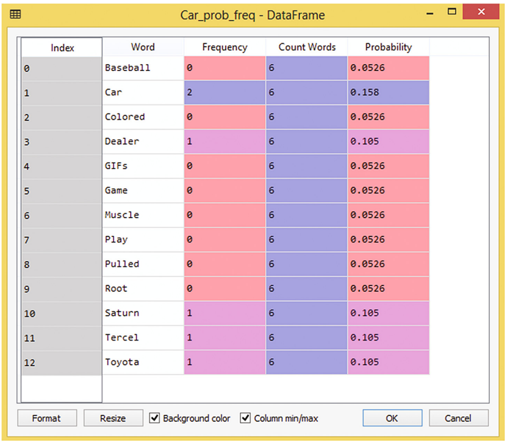
So,

$$P(Computer/S1) = 0.2 * 0.0625 * 0.0625 * 0.0625 = 0.000048828$$

So as the probability of the given input among Game class is high, this text belongs to the Game class.

Similarly, the probability calculations for the Game Class and Computer Class has been performed and the results obtained are [0.0526, 0.105, 0.105, 0.105, 0.150, 0.0526, 0.0526, 0.0526, 0.0526, 0.0526, 0.105, 0.0526, 0.0526] and [0.0625, 0.0625, 0.0625, 0.0625, 0.0625, 0.125, 0.0625, 0.0625, 0.125, 0.125, 0.0625, 0.0625, 0.0625] respectively.

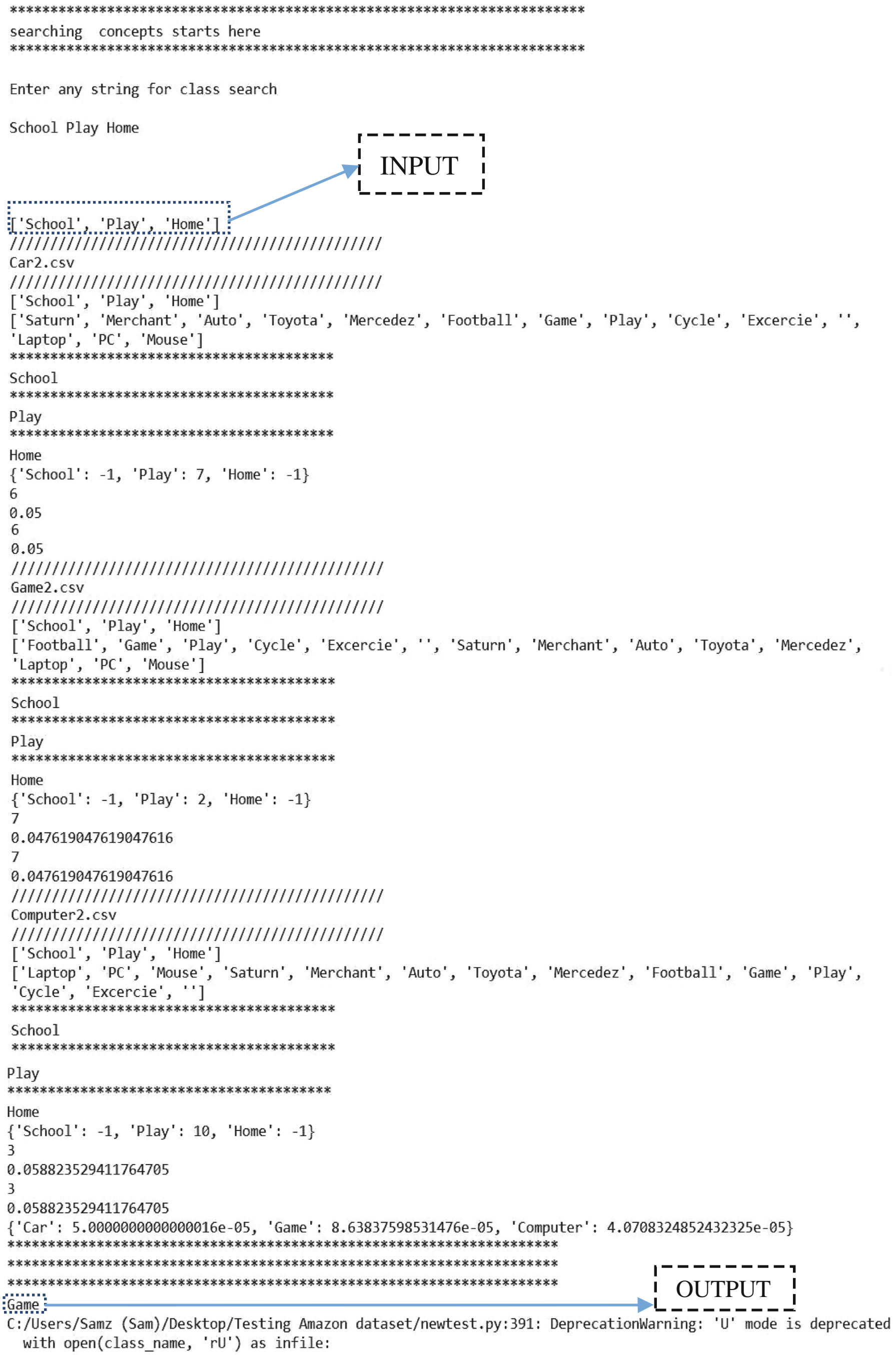
Figure 6 shows the final calculation result of the considered input set. The corresponding algorithms are given in Table 1.



Index	Word	Frequency	Count Words	Probability
0	Baseball	0	6	0.0526
1	Car	2	6	0.158
2	Colored	0	6	0.0526
3	Dealer	1	6	0.105
4	GIFs	0	6	0.0526
5	Game	0	6	0.0526
6	Muscle	0	6	0.0526
7	Play	0	6	0.0526
8	Pulled	0	6	0.0526
9	Root	0	6	0.0526
10	Saturn	1	6	0.105
11	Tercel	1	6	0.105
12	Toyota	1	6	0.105

Fig. 5. Car class with their frequency, count words (total words in car class), probability from the catalog in Fig. 3

5 Results and Conclusion



```
************************************************************************
searching  concepts starts here
************************************************************************

Enter any string for class search

School Play Home



['School', 'Play', 'Home']
///////////////////////////////////////////////
Car2.csv
///////////////////////////////////////////////
['School', 'Play', 'Home']
['Saturn', 'Merchant', 'Auto', 'Toyota', 'Mercedez', 'Football', 'Game', 'Play', 'Cycle', 'Excercie', '',
'Laptop', 'PC', 'Mouse']
****************************************
School
****************************************
Play
****************************************
Home
{'School': -1, 'Play': 7, 'Home': -1}
6
0.05
6
0.05
///////////////////////////////////////////////
Game2.csv
///////////////////////////////////////////////
['School', 'Play', 'Home']
['Football', 'Game', 'Play', 'Cycle', 'Excercie', '', 'Saturn', 'Merchant', 'Auto', 'Toyota', 'Mercedez',
'Laptop', 'PC', 'Mouse']
****************************************
School
****************************************
Play
****************************************
Home
{'School': -1, 'Play': 2, 'Home': -1}
7
0.047619047619047616
7
0.047619047619047616
///////////////////////////////////////////////
Computer2.csv
///////////////////////////////////////////////
['School', 'Play', 'Home']
['Laptop', 'PC', 'Mouse', 'Saturn', 'Merchant', 'Auto', 'Toyota', 'Mercedez', 'Football', 'Game', 'Play',
'Cycle', 'Excercie', '']
****************************************
School
****************************************

Play
****************************************
Home
{'School': -1, 'Play': 10, 'Home': -1}
3
0.058823529411764705
3
0.058823529411764705
{'Car': 5.0000000000000016e-05, 'Game': 8.63837598531476e-05, 'Computer': 4.0708324852432325e-05}
********************************************************************
********************************************************************
********************************************************************
Game
C:/Users/Samz (Sam)/Desktop/Testing Amazon dataset/newtest.py:391: DeprecationWarning: 'U' mode is deprecated
  with open(class_name, 'rU') as infile:
```

Fig. 6. Calculation result of above input

Table 1. Classification algorithm

Training and Applying Multinomial Naïve Bayes Algorithm

```
(a) TRAINMULTINOMIALNAIYEBAYES(C, F)

    P $\leftarrow$ FINDVOCABULARY (F)
    N $\leftarrow$ COUNTDOCUMENTS (F)
    for each c Є C
    do K $\leftarrow$ COUNTDOCUMENTSINCLASS (F, c)
        prior[c] $\leftarrow$ $K_c/K$
        $word_c$ $\leftarrow$ CONCATTEXTOFALLDOCUMENTSINCALSS (F, c)
        for each j Є P
        do $J_{cj}$ $\leftarrow$ COUNTTOKENSOFTERM ($word_c$ j)
        for each j Є P
        do condprob[j][c] $\leftarrow$ $\frac{J_{cj}+1}{\sum_{j`}(J_{cj`}+1)}$
        return P, prior, condprob

(b) APPLYMULTINOMIALNAIYEBAYES (C, P, prior, condprob, d)
    T $\leftarrow$ EXTRACTWORDTOKENFROMDOCS (P, d)
    for each i Є C
    do score[i] $\leftarrow$ log prior[i]
        for each j Є T
        do score[i] += log condprob[j][i]
    return arg $\max_{i \in C}$ score[i]
```

Our experiments are carried out from product databases of Amazon, Flipkart, Snapdeal and Paytm. The database currently contains 40000 product catalogs and classification structure contains 1000 leaf classes. This experiment has been carried out on Intel Core i3 1.80 GHz machine which has 4 GB of RAM. Database server used is MySQL and the application software for implementation of programming code is Anaconda (Spyder 3.6). Approximately 70% accurate results were obtained based on our algorithm and it manages attribute-wise distribution of terms to adapt to the organized way of e-lists. The best thing is that with the help of normalization, our method without giving weightage to long text is able to give better results. We are in the process of improving the accuracy hence obtained. The algorithm could be made more powerful by including information from more sources. Test information drawn from a more extensive source would likewise give a superior speculation estimate.

References

1. Fensel, D., Ding, Y., Schulten, E., Omelayenko, B., Botquin, G., Brown, M., Flett, A.: Product data integeration in B2B e-commerce. IEEE Intell. Sys. **16**(3), 54–59 (2001)
2. Ding, Y., Korotkiy, M., Omelayenko, B., Kartseva, V., Zykov, V., Klein, M., Schulten, E., Fensel, D.: GoldenBullet: automated classification of product data in e-commerce. In: Business Information System (2002)
3. Branwen, G.: Silk road: theory and practice (2015). http://www.gwern.net/Silk%20Road, Accessed 09 Dec 2015
4. Denoyer, L., Gallinari, P.: Bayesian network model for semi-structured document classification. Inf. Process. Manag. (Elsevier) **40**(5), 807–827 (2004)
5. Zaki, M.J., Aggarwal, C.C.: XRules: an effective structural classifier for XML data. In: 9th ACM SIGKDD, pp. 316–325 (2003)
6. Denoyer, L., Vittaut, J., Gallinari, P., Brunessaux, S., Brunessaux, S.: Structured multimedia document classification. In: ACM DOCENG 2003, pp. 153–160 (2003)
7. Vinithra, S.N., Anand Kumar, M, Soman, K.P.: Analysis of sentiment classification for Hindi movie reviews: a comparison of different classifiers. Int. J. Appl. Eng. Res. **10** (2015)
8. Ani, R., Sasi, G., Sankar, U.R., Deepa, O.S.: Decision support system for diagnosis and prediction of chronic renal failure using random subspace classification. In: International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1287–1292. Jaipur (2016)
9. Priyanka, C., Gupta, D.: Fine grained sentiment classification of customer reviews using computational intelligent technique. Int. J. Eng. Technol. **7**(4), 1453–1468 (2015)
10. Sebastiani, F.: Machine learning in automated text categorization. ACM Comput. Surv. **34**(1), 1–47 (2002)
11. Mitchell, T.: Machine Learning. McGraw-Hill, Columbus (1997)