Asian Journal of Statistical Sciences
Vol. 1, No-2, (2021), pp. 159-172
© ARF India, Gurgaon, India

ARF INDIA
Academic Open Access Publishing
Journal homepage link:
http://arfjournals.com/ajss

Supervised Multinomial Text Topic Identification using Naïve Bayes

Kanchan Jain[a], Suresh Kumar Sharma[b] and Gurpreet Singh Bawa[c]

[a] *Department of Statistics, Panjab University, Chandigarh, India.* [b] *Department of Statistics, Panjab University, Chandigarh, India.* [c] *Accenture Chicago Innovation Hub, Madison St, Chicago, IL 60661, United States.*

ARTICLE HISTORY
Compiled December 12, 2021

Received 06 June 2021; Accepted 24 November 2021

ABSTRACT
In this paper, the focus is on Multinomial document model which is similar to the Bernoulli model, but the presence flag in the former is replaced with the frequentist method which takes into account the number of times the tokens occur in the text. The application of Naïve Bayes approach is discussed for the document models. Estimators in Naïve Bayes and Multinomial setup have been derived. Illustration and R code snippets for implementation are included.

KEYWORDS
Sections; lists; figures; tables; references; appendices

1. Introduction

Content sharing sites are increasingly becoming a source of vital behavioural and attitudinal information for many organizations. Corporations are realizing the economic value of the information stored in such sites (Ghose and Panagiotos, 2010). Blog sites, online journals, wiki pages and social media sites have tons of information created by users to express their varied opinions from the launch of latest tech-loaded smartphones to political happenings and to soccer league results. On micro-blogging sites such as Twitter, there are around 330 million active users creating about 500 million tweets per day. An important aspect of data on these social networks is that people active on these platforms create data instantaneously in real time. Since users are real and not anonymous, they give a fair idea about people's opinion and attitudes towards a brand, firm or service. For corporations, it also gives a way to assess drivers for future sales and business (Dhar and Chang, 2009). People from almost every age-group express themselves on social media. About 75% of the internet users use social media and this number is increasing day by day (Kaplan and Haenlein, 2010).
Data generated from online sources are relatively easier to acquire and act as potential treasure of information for discovering insights (Dey and Haque, 2008). Extracting this information leads to uncovering of valuable insights in the fields of marketing, services, human resources and customer relationship management. Hence more and more orga-

CONTACT Kanchan Jain[a]. Email: jaink14@gmail.com

nizations are focusing on social media to realize their organizational goals (Murdough, 2009).

A statistical analysis on social media was carried out by Dhar and Chang (2009). They conducted statistical regression and correlation analysis on social unstructured data to forecast the sales in the music industry. A study was conducted on influence of social media in the financial markets by Tirunillai and Tellis (2012) who wanted to establish links between crowdsourced information in social media to the stock prices on the exchanges.

Big Data are data having a lot of variety and high volume and is being produced at a very high rate and is incapable of being collected, handled, and analyzed using prevailing statistical models, traditional tools and architectures. Big Data are of three types: Structured; Semi-structured and Unstructured. Structured data contributes to merely 20% of the existing data and exist as relational databases, tabular sheets, files etc. Unstructured data does not have any fixed structural formats or schema. Examples include text, audio, images and video or any other form of information that does not fit the common tabular notion of datasets. For example, blog posts, online forum discussions, social media data, email content, web-page content, audio data, video streams, image data etc. E-commerce web platforms, social data and sensors from intelligent devices are significantly contributing to unstructured data. Semi- structured data sources have no strict standard of formatting and include XML files, sensor logs and web logs etc.

Unstructured textual data comes from a variety of sources like documents, emails, online forums, electronic news content, blogs, social media feeds and posts, call center logs, customer feedback etc. Text analytics aim to retrieve information and extract value from the text-based datasets. Three pillars primarily supporting the idea of Text Analytics are-Statistical Analysis, Computational Linguistics and Machine Learning/ Deep Learning.

Unstructured textual data from social media, websites, emails etc. has to be processed to be structurally suitable for analysis (Rusu, Halcu, Grigoriu, Neculoiu, Sandulescu and Marinescu, 2013). Text analytics have a wide variety of applications in decision making and strategy. Quantification of text data or other approaches to bring structural stability can be performed in a number of ways. Some approaches are discussed below:

Binary approach: Suppose there is a set of documents, say social media posts, and a word of interest, say ‘iPhone’. One very simple way to structure the data is to create a flag for each record in the dataset on whether word of interest is present or not. The above random variable of interest follows a Bernoulli Distribution where X represents the presence/absence of a characteristic. For example, the comment ‘Iphone is too costly. I don’t want to buy an iphone’ gets the flag 1 and ‘I am going for the concert today’ gets flag 0.

Frequentist approach: Suppose the same set of documents as above and the same word of interest - ‘iPhone’ is considered. Another way to structure data is to calculate frequencies for each record in the dataset based on the word of interest. If X denotes the number of occurrences of an event, then X follows Poisson Distribution. Using this approach, the comment ‘Iphone is too costly. I don’t want to buy an iphone’ gets value 2, ‘The new iphone has smooth User Interface’ has value 1 and ‘I am going for the concert today’ is taken as 0.

In this paper, a Multinomial distribution of the words present in the labelled documents (also known as the training dataset) is used. In this model setup, the feature vectors of the document inherently capture the word frequencies (Manning, Raghavan and Schutze, 2008), and not merely their occurrence in that document, as in Bernoulli Model. Additionally, Multinomial Naïve Bayes setup has been used for deriving estimates of probabilities. The problem faced due to zero probability has been stated and Laplace rule of succession discussed for addressing the problem.

In Section 2, the multinomial document model setup is discussed. This is followed by Section 3 which discusses the Naïve Bayes setup. In Section 4, the MLE for Multinomial distribution is derived. An example for the same has been provided in Section 5. Section 6 deals with the problem of zero-probability. Implementation of a real-life problem in R and Python is included in this section. Conclusions are reported in Section 7. Appendix consists of snapshots of R and Python codes.

2. Multinomial Document Model Setup

Let X be a multinomial model document feature vector for D.

x_t, the t^{th} component of X, gives the frequency of the word w_t occurring in document D, $n = \sum_t x_t$ is total count of words in document D,

C_k is document belonging to category or topic k.

We assume that $P(w_t|C_k)$ denotes probability of word wt being present in document belonging to category k. The estimation is being done with help of word count data from feature vectors of the document. Naïve Bayes assumptions state that the words present in documents exhibit independence among themselves. The document likelihood $P(D|C_k)$ can then be written in the form of a multinomial distribution, in which the count of draws represents the count of words in the specific document. $P(w_t|C_k)$ is the probability of word t being present in document belonging to category k.

$$\begin{aligned} P(D|C_k) = P(X|C_k) &= \frac{n!}{\prod_{t=1}^{|V|} x_t!} \prod_{t=1}^{|V|} P(w_t|C_k)^{x_t} \\ &\propto \prod_{t=1}^{|V|} P(w_t|C_k)^{x_t} \end{aligned} \tag{1}$$

With respect to the multinomial model, the likelihood parameters $P(w_t|C_k)$ are probabilities of occurrence of every word conditional on the document category or topic. In addition to that, parameters of the model are inclusive of prior probabilities $P(C_k)$. For estimating the parameters based on the training document dataset $\{D_1, ..., D_N\}$, let Z_{ik} be a variable which assumes value 1 if D_i belongs to category or topic k, and 0 otherwise where N is the total count of documents and x_{it}, is frequency of word wt in document D_i, calculated for each word w_t in vocabulary V. The estimated probabilities are written as

$$\hat{P}(w_t|C_k) = \frac{\sum_{t=1}^{N} x_{it}\ Z_{ik}}{\sum_{s=1}^{|V|} \sum_{i=1}^{N} x_{is} Z_{ik}} = \frac{n_k(w_t)}{\sum_{i=1}^{|V|} n_k(w_s)} \tag{2}$$

where $n_k(w_t)$ denotes word frequency for w_t in the k^{th} category/topic documents and $|V|$ is the number of words in vocabulary V.

(2) estimates $P(w_t|C_k)$ as the relative frequency of words w_t against the total count

of words in the document belonging to category or topic k, where

$$n_k(w_t) = \sum_{i=1}^{N} x_{it}\ Z_{ik}^2$$

The prior probability of k^{th} category or topic can be estimated as

$$P(\hat{C_k}) = \frac{N_k}{N} \tag{3}$$

where N_k denotes the count of documents belonging to k^{th} category or topic and N is the total count of documents in the training dataset.
Hence, for a given training dataset containing documents associated with a category or topic label and a set of k such categories or topics, a multinomial classification model can be estimated in the following manner:

(1) Start off with defining the vocabulary V where the total count of words present specifies the feature vector dimensionality,
(2) Count N, N_k and x_{it} ,
(3) Estimate $P(w_t|C_k)$ using (2),
(4) Finally, priors $P(C_k)$ to be estimated using (3).

Thus, classifying an unseen document D would require estimating posterior probability of each category or topic as

$$P(C_k|D) = P(C_k|X) \propto P(X|C_k)\ P(C_k) \propto P(C_k)\ \prod_{t=1}^{|V|} P(w_t|C_k)^{x_t}$$

In contrast to the Bernoulli model, here words that are absent in the concerning document (that is, words for which $x_t = 0$) have no effect on the probability. Using words u occurring in the document, the posterior probability can be written

$$P(C_k|D) \propto P(C_k)\ \prod_{j=1}^{len(D)} P(u_j|C_k)$$

where u_j denotes j^{th} word of D^{th} document and $len(D)$ represents length of document D.

3. Naïve Bayes Model and Maximum Likelihood Estimation

Given a training dataset $(x^{(i)}, y^{(i)})$ where each $x^{(i)}$ is a vector for i from 1 to n and $y^{(i)}$ ranges from 1 to k, where k is an integer that specifies the associated count of categories or topics in question. Essentially, the task in question is a multi-category classification task, with objective of mapping every input realization x to one of the categories that y is capable of assuming out of k possible choices. If $k = 2$, the associated task translates to a binary classification task.
For simplicity, assume that every x belongs to the set $\{-1, +1\}^d$ for a value of d

which specifies the count of model "features". It can be alternately stated that every component x_j, for j ranging from 1 to d, is capable of assuming either of the two given values.
The Naïve Bayes (NB) model is explained below: (Qin, Tang and Chen, 2012).
Let there be random variables Y and $X_1 ... X_d$ associated to category label y and the vector components x_i. Then for every label y in pair with attributes of $x_1 ... x_d$.

$$P(Y = y, X_1 = x_1, X_2 = x_2, \ldots, X_d = x_d) = P(Y = y) \prod_{j=1}^{d} P(X_j = x_j | Y = y) \quad (4)$$

The naivety in (4) is because the NB assumption is relatively strong one. However, it is an extremely useful one since it brings a dramatic reduction in the count of model parameters, even though the model in question practically stays quite effective.
From (3), the parameters of the model are of two kinds

$$q(y) = P(Y = y) \qquad for\ every\ y \in \{1 \ldots k\} \quad (5)$$
$$q_j(x|y) = P(X_j = x|Y = y) \quad (6)$$

Finally, the probability for any y, $x_1 ... x_d$ can be written as:

$$P(y \cap x_1 \cap x_2 \cap \ldots \cap x_d) = q(y) \prod_{j=1}^{d} q_j(x_j|y) \quad (7)$$

For an unseen test realization $x = \{x_1, x_2, ..., x_d\}$, we need to maximize (4) to obtain estimates of (5) and (7).
For the parameters in Section 3, the Maximum Likelihood Estimate of (5) can be derived as

$$\hat{q}(y) = \frac{\sum_{i=1}^{n} [[y^{(i)} = y]]}{n} = \frac{count(y)}{n} \quad (8)$$

In (8), $[[y(i) = y]]$ takes value 1 if $y^{(i)} = y$ and otherwise, it is 0. Similarly, the MLE for (7) can be derived as:

$$\hat{q}_j(x|y) = \frac{\sum_{i=1}^{n} [[y^i = y \cap x_j^i = x]]}{\sum_{i=1}^{n} [[y^i = y]]} = \frac{count_j(x|y)}{count(y)} \quad (9)$$

where $count_j(x|y) = \sum_{i=1}^{n} [[y^i = y \cap x_j^i = x]]$.

4. Example of Multinomial Document Modeling

Let there be a collection of documents, every one of which belongs to one of the two topics: Sports or Informatics, denoted by S and I respectively. For a given training dataset having eleven documents with six for Sports and five for informatics, the objective is estimation for a Naive Bayes classifier on basis of the Multinomial model and to label unseen documents pertaining to Sports or Informatics.

Let the vocabulary V consist of 8 words given by

$$\begin{bmatrix} w_1 = goal \\ w_2 = tutor \\ w_3 = variance \\ w_4 = speed \\ w_5 = drink \\ w_6 = defence \\ w_7 = performance \\ w_8 = field \end{bmatrix}$$

A document D_i can now be denoted as a row vector m_i where m_{it} denotes the count of word w_t in D_i. Training dataset is shown below in form of a matrix for each category or topic where each row signifies a document vector of 8 dimensions.

$$M^{Sport} = \begin{bmatrix} 2 & 0 & 0 & 0 & 1 & 2 & 3 & 1 \\ 0 & 0 & 1 & 0 & 2 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 2 & 1 & 0 \\ 1 & 0 & 0 & 2 & 0 & 1 & 0 & 1 \\ 2 & 0 & 0 & 0 & 1 & 0 & 1 & 3 \\ 0 & 0 & 1 & 2 & 0 & 0 & 2 & 1 \end{bmatrix}$$

$$M^{Inf} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 2 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

Suppose that the objective is to categorize the following test documents

$$D_1 = w_5 w_1 w_6 w_8 w_1 w_2 w_6$$

$$D_2 = w_3 w_5 w_2 w_7$$

into either Sports or Informatics.

Denoting the word frequency of w in every document of topic k (S or I) by $n_k(w)$,

$$\hat{P}(w|S) = \frac{n_s(w)}{\sum\limits_{v \in V} n_s(v)}$$

,

$$\hat{P}(w|I) = \frac{n_I(w)}{\sum\limits_{v \in V} n_I(v)}.$$

The estimates of probabilities of different words in V (vocabulary) are given in Table 1 where second and fourth columns give the number of words in Sports (S) and Informatics (I) categories.

Table 1. *Estimates of Probabilities of Words in Vocabulary*

w	$n_s(w)$	$\hat{P}(w\|S)$	$n_I(w)$	$\hat{P}(w\|I)$
w_1	5	$\frac{5}{36}$	1	$\frac{1}{16}$
w_2	1	$\frac{1}{36}$	4	$\frac{4}{16}$
w_3	2	$\frac{2}{36}$	3	$\frac{3}{16}$
w_4	5	$\frac{5}{36}$	1	$\frac{1}{16}$
w_5	4	$\frac{4}{36}$	1	$\frac{1}{16}$
w_6	6	$\frac{6}{36}$	2	$\frac{2}{16}$
w_7	7	$\frac{7}{36}$	3	$\frac{3}{16}$
w_8	6	$\frac{6}{36}$	1	$\frac{1}{16}$

Using estimated probabilities from Table 1, estimates of posterior probabilities for D_1 and D_2 are calculated as follows:

$$\begin{aligned}
\hat{P}(D_1|S) &= \hat{P}(w_5|S)\ \hat{P}(w_1|S)\ \hat{P}(w_6|S)\ \hat{P}(w_8|S)\ \hat{P}(w_1|S)\ \hat{P}(w_2|S)\ \hat{P}(w_6|S) \\
&= \frac{4}{36}\ \frac{5}{36}\ \frac{6}{36}\ \frac{6}{36}\ \frac{5}{36}\ \frac{1}{36}\ \frac{6}{36} \\
&= \frac{4 \times 5^2 \times 6^3}{36^7} = 2.76 \times 10^{-7} \\
\hat{P}(D_1|I) &= \hat{P}(w_5|I)\ \hat{P}(w_1|I)\ \hat{P}(w_6|I)\ \hat{P}(w_8|I)\ \hat{P}(w_1|I)\ \hat{P}(w_2|I)\ \hat{P}(w_6|I) \\
&= 5.96 \times 10^{-7} \\
\hat{P}(S|D_1) &= \frac{\hat{P}(S)\ P(D_1|S)}{\hat{P}(S)\ \hat{P}(D_1|S) + \hat{P}(I)\ \hat{P}(D_1|I)} \\
&= \frac{1.50 \times 10^{-7}}{1.50 \times 10^{-7} + 2.71 \times 10^{-8}} \approx 0.847 \\
\hat{P}(I|D_1) &= 1 - \hat{P}(S|D_1) \approx 0.153
\end{aligned}$$

Since $\hat{P}(S|D_1) > \hat{P}(I|D_1)$, hence we classify D_1 as S. Similarly, for D_2 :

$$\begin{aligned}
\hat{P}(D_2|S) &= \hat{P}(w_3|S)\ \hat{P}(w_5|S)\ \hat{P}(w_2|S)\ \hat{P}(w_7|S)\\
&= \frac{2}{36}\ \frac{4}{36}\ \frac{1}{36}\ \frac{7}{36}\\
&= \frac{2^3 \times 7}{36^4} \approx 3.33 \times 10^{-5}\\
\hat{P}(D_2|I) &= \hat{P}(w_3|I)\ \hat{P}(w_5|I)\ \hat{P}(w_2|I)\ \hat{P}(w_7|I)\\
&= \frac{3}{16} \times \frac{3}{16} \times \frac{3}{16} \times \frac{3}{16} = 5.9 \times 10^{-4}
\end{aligned}$$

Posterior probabilities are calculated as below:

$$\begin{aligned}
\hat{P}(S|D_2) &= \frac{\hat{P}(S)\ P(D_2|S)}{\hat{P}(S)\ \hat{P}(D_2|S) + \hat{P}(I)\ \hat{P}(D_2|I)}\\
&= \frac{1.82 \times 10^{-5}}{1.82 \times 10^{-5} + 2.50 \times 10^{-4}} \approx 0.0679\\
\hat{P}(I|D_2) &= 1 - \hat{P}(S|D_2) \approx 0.932
\end{aligned}$$

5. Problem of Zero-Probability

A primary shortcoming of relative frequency estimation with respect to the multinomial model is the resultant estimation of zero probability due to zero counts. It is not favorable as the following likelihood equation

$$\begin{aligned}
P(D|C) = P(X|C_k) &= \frac{n!}{\prod_{t=1}^{|V|} x_i} \prod_{t=1}^{|V|} P(w_t|C_k)^{x_i}\\
&\propto \prod_{t=1}^{|V|} P(w_t|C_k)^{x_i}
\end{aligned}$$

takes product of associated probabilities. Now in the product, the presence of any one zero term renders the whole product as zero. As a consequence, the probability of the document being associated to that specific topic or class equals zero.

Intuitively, it would mean that since that particular word is absent in the document topic or category in the given training dataset, it would not be present in any document belonging to that category, which is not true intuitively.

The complex part is that

$$\hat{P}(w_t|C_k) = \frac{\sum_{i=1}^{N} x_{it} \times Z_{ik}}{\sum_{s=1}^{|V|} \sum_{i=1}^{N} x_{is} \times Z_{ik}} = \frac{n_k(w_t)}{\sum_{i=1}^{|V|} n_k(w_s)}, \tag{10}$$

happens to cause an underestimation of the likelihood estimates of those words that are absent in the training dataset. Even if word w is absent for the k^{th} topic or category in the training dataset, it still needs to have $P(w|C_k) > 0$. Also, since sum of probabilities is equal to 1, unobserved words have underestimated probabilities and present words

have some overestimation.
Hence, an approach for alleviating the issue involves removal of a trivial probability assigned to observed words and distributing the amount among the unseen words. There is a method of doing the redistribution, which is often known as Laplace's law of succession (Laplace, 1814). It is also referred to as add-one smoothing (Raman, 2000) since it effectively does an addition of a count of 1 against each type of word. For instance, if the training dataset consists of W terms, then (10) can be replaced with:

$$\begin{aligned}\hat{P}_{Lap}(w_t|C_k) &= \frac{1+\sum_{i=1}^{N} x_{it}\times Z_{ik}}{|V|+\sum_{i=1}^{|V|}\sum_{i=1}^{N} x_{is}\times Z_{ik}}\\ &= \frac{1+n_k(w_t)}{|V|+\sum_{i=1}^{|V|} n_k(w_s)}\end{aligned}$$

It can be seen that denominator has been inflated to account for the $|V|$ additional "observations" which resulted due to the "add 1" component, thereby making sure that the probabilities remain normalized.

6. Real-life Implementation of Multinomial Setup in R

- In this section, implementation of Multinomial Naïve Bayes on text data is depicted using R Software. Data being used is Polarity dataset v2, a movie reviews dataset, Cornell IMDb available on https://www.cs.cornell.edu/people/pabo/movie-review-data/ and containing reviews of 2000 movies with an associated positive or negative label based on the sentiment using the queries.
  Sequence of steps is as given below:

- required packages are imported into the R environment;

- snapshot of the dataset can be seen by clicking on the object in the environment;

- ordering of the dataset is randomized to remove any sort of patterns in labels while getting the train and test split and can again be checked in the R environment by clicking on the object;

- the 'class' variable is converted to type 'factor' and a corpus of the 'text' variable is created;

- corpus cleaned by data pre-processing – conversion to lower case, removing punctuations, removing numbers, removing stop words and clearing leading white spaces;

- a document term matrix (DTM) is created on the clean corpus;

- On inspecting the DTM, the following is observed:

Figure A1 : **Resultant DTM built on Cleaned Corpus**

- number of terms (38957), which can be features, is reduced by considering those terms which occur at least in 5 or more documents for the analysis. Thus, a dictionary of only those terms is created and DTM is constructed on that basis, using queries in Figure 1;

- Since the dataset was already randomized, the first 1500 rows are taken as training set and remaining 500 rows as test set ;

- DTM objects are converted back to data frames for ease of handling;

- Finally, Multinomial Naïve Bayes model is fitted on the training dataset and used to make predictions on the test set.

Figure A2 : **Fitting Model and Predicting**

- Once the model has been fitted, the truth table of predictions vs actuals is observed.

Figure A3 : **Truth Table of Predicted vs Observed**

- Confusion Matrix function is used to build the confusion matrix and get relevant statistics.

Figure A4 : **Confusion Matrix and Relevant Statistics**

In Figure 4, the fitted model shows an accuracy of 57% in classifying the unlabeled observations. Other model diagnostics are as given below:

- **Confidence Interval (CI):** (0.5273, 0.6158).
- **No Information Rate,** the finest approximation conditional that zero information beyond the complete class distribution is provided, is 0.508.
- **Kappa (Cohen, 1960):** In our case, Kappa takes the value 0.1534, which lies in the poor range.
- **Mcnemar's Test P-Value** (McNemar, 1947): For our example, it equals 2.2×10^{-6} which shows an association between dependent and independent variable.
- **Sensitivity** of 0.9268 is very good.
- **Specificity** of 22.83%which is moderate.
- **Predicted Values:** Positive Predicted Value (PPV) and Negative Predicted Value (NPV) are 0.5377 and 0.7632 respectively.
- **Prevalence** of 0.4920 indicates share of cases in the given population at an instance.

The values of model diagnostics show that the fitted model is classifying the documents moderately. **However, this conclusion is case-specific and cannot be generalized.**

The same implementation can be done in Python too and the code is shown below:

Figure A5 : Python Code

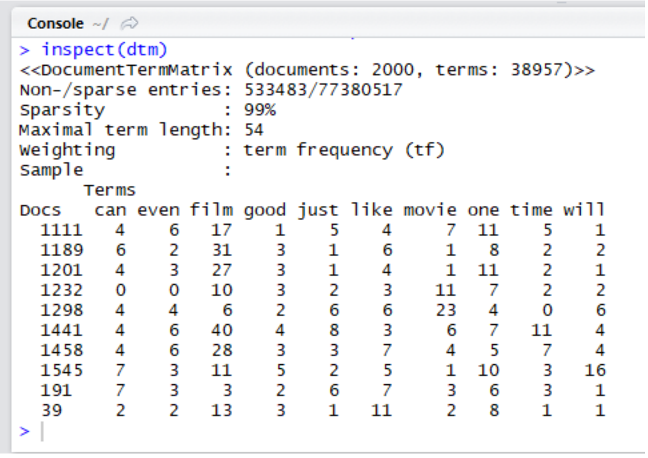
7. Conclusions

In this paper, categorization of unstructured documents is done using underlying Multinomial distribution for words. The posterior probabilities are obtained from the training dataset which is pre-labeled (Supervised model). This approach is lexical in nature and focuses on the frequency of the words and not just their presence/absence across the documents. Categorization has been done for Multinomial document classification model by using estimates of probabilities. In real life illustration, eleven documents (six belonging to Sports category and five belonging to Informatics) have been considered and eight words chosen in the vocabulary. It has been explained how to categorize two new documents based on these words. Movie reviews dataset, Polarity dataset v2 containing reviews of 2000 movies has been analyzed.

Acknowledgement

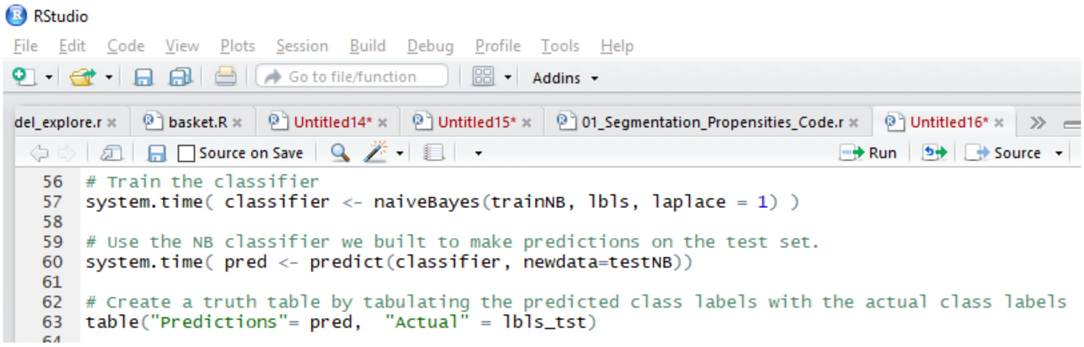
The authors are extremely grateful to the editor and the referees for their valuable suggestions which helped in improving the manuscript

Appendix A.



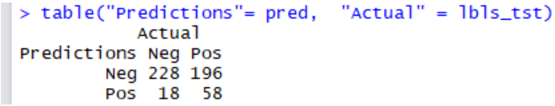
```
> inspect(dtm)
<<DocumentTermMatrix (documents: 2000, terms: 38957)>>
Non-/sparse entries: 533483/77380517
Sparsity           : 99%
Maximal term length: 54
Weighting          : term frequency (tf)
Sample             :
      Terms
Docs   can even film good just like movie one time will
  1111   4    6   17    1    5    4     7  11    5    1
  1189   6    2   31    3    1    6     1   8    2    2
  1201   4    3   27    3    1    4     1  11    2    1
  1232   0    0   10    3    2    3    11   7    2    2
  1298   4    4    6    2    6    6    23   4    0    6
  1441   4    6   40    4    8    3     6   7   11    4
  1458   4    6   28    3    3    7     4   5    7    4
  1545   7    3   11    5    2    5     1  10    3   16
  191    7    3    3    2    6    7     3   6    3    1
  39     2    2   13    3    1   11     2   8    1    1
> 
```

Figure A1. Resultant DTM built on Cleaned Corpus



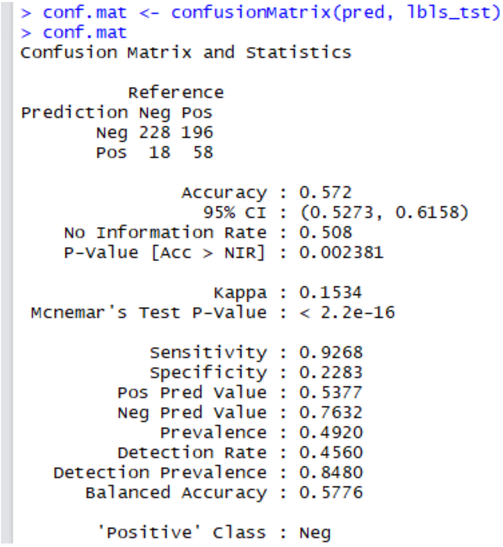
```
# Train the classifier
system.time( classifier <- naiveBayes(trainNB, lbls, laplace = 1) )

# Use the NB classifier we built to make predictions on the test set.
system.time( pred <- predict(classifier, newdata=testNB))

# Create a truth table by tabulating the predicted class labels with the actual class labels
table("Predictions"= pred,  "Actual" = lbls_tst)
```

Figure A2. Fitting Model and Predicting



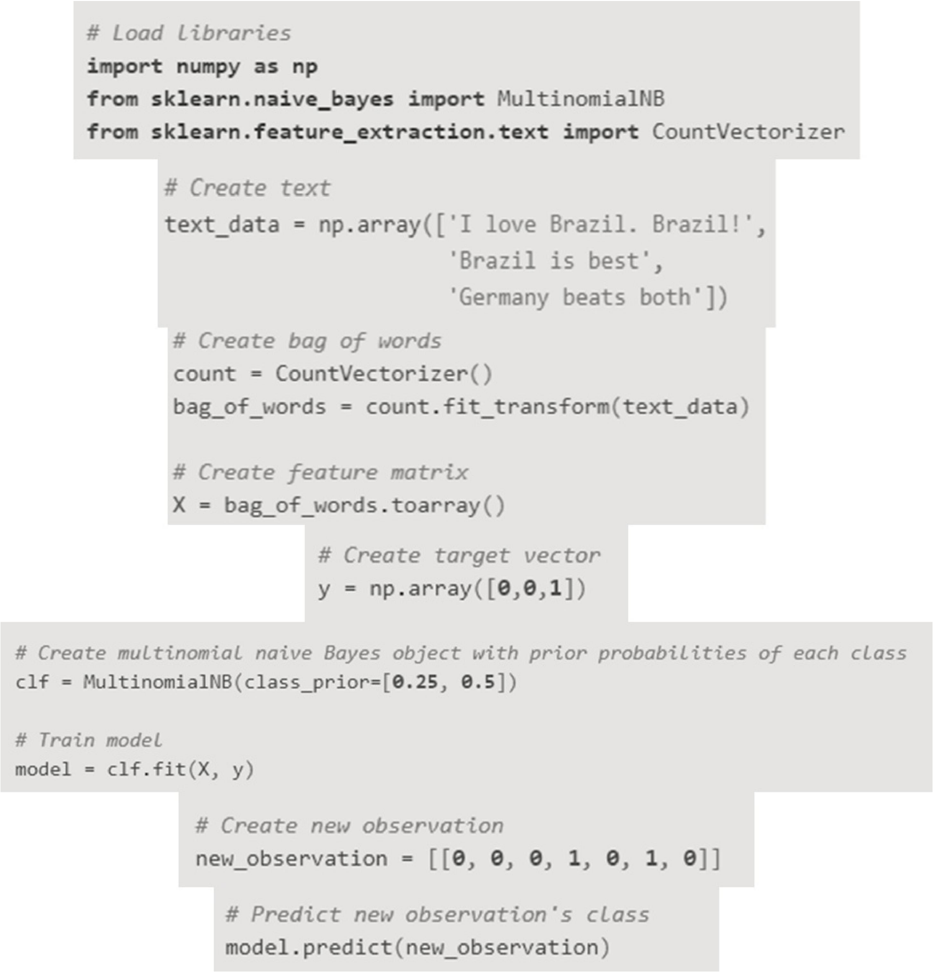
```
> table("Predictions"= pred,  "Actual" = lbls_tst)
           Actual
Predictions Neg Pos
        Neg 228 196
        Pos  18  58
```

Figure A3. Truth Table of Predicted vs Observed



```
> conf.mat <- confusionMatrix(pred, lbls_tst)
> conf.mat
Confusion Matrix and Statistics

          Reference
Prediction Neg Pos
       Neg 228 196
       Pos  18  58

               Accuracy : 0.572
                 95% CI : (0.5273, 0.6158)
    No Information Rate : 0.508
    P-Value [Acc > NIR] : 0.002381

                  Kappa : 0.1534
 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9268
            Specificity : 0.2283
         Pos Pred Value : 0.5377
         Neg Pred Value : 0.7632
             Prevalence : 0.4920
         Detection Rate : 0.4560
   Detection Prevalence : 0.8480
      Balanced Accuracy : 0.5776

       'Positive' Class : Neg
```

Figure A4. Confusion Matrix and Relevant Statistics



```python
# Load libraries
import numpy as np
from sklearn.naive_bayes import MultinomialNB
from sklearn.feature_extraction.text import CountVectorizer

# Create text
text_data = np.array(['I love Brazil. Brazil!',
                      'Brazil is best',
                      'Germany beats both'])

# Create bag of words
count = CountVectorizer()
bag_of_words = count.fit_transform(text_data)

# Create feature matrix
X = bag_of_words.toarray()

# Create target vector
y = np.array([0,0,1])

# Create multinomial naive Bayes object with prior probabilities of each class
clf = MultinomialNB(class_prior=[0.25, 0.5])

# Train model
model = clf.fit(X, y)

# Create new observation
new_observation = [[0, 0, 0, 1, 0, 1, 0]]

# Predict new observation's class
model.predict(new_observation)
```

Figure A5. Python Code

References

[1] Cohen, J. (1960). A coefficient of agreement for nominal scales, Educational and Psychological Measurement, 20 (1), 37–46.

[2] Dey, L. and Haque, S. M. (2008). Opinion mining from noisy text data, In Proceedings of the second workshop on analytics for noisy unstructured text data, 83–90.

[3] Dhar, V. and Chang, E. A. (2009). Does chatter matter? the impact of user-generated content on music sales, Journal of Interactive Marketing, 23(4), 300–307.

[4] Ghose, A. and Panagiotos, I. (2010). The economining project at nyu: Studying the economic value of user-generated content on the internet, Journal of Revenue and Pricing Management, 8, 241–246.

[5] Kaplan, A. M. and Haenlein, M. (2010). Users of the world, unite! the challenges and opportunities of social media, Business Horizons, 53(1), 59–68.

[6] Laplace, P. S. (1814). Essai philosophique sur les probabilities, Courtier, Paris.

[7] Manning, D. C., Raghavan, P. and Schutze, H. (2008). Introduction to Information Retrieval, Cambridge University Press, 253 – 280.

[8] McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages, Psychometrika, 12 (2), 153–157.

[9] Murdough, C. (2009). Social media measurement: It's not impossible, Journal of Interactive Advertising, 10, 94–99.

[10] Qin, F., Tang, X. and Cheng, Z. (2012). Application and research of multi-label Naïve Bayes Classifier, Proceedings of the 10th World Congress on Intelligent Control and Automation, Beijing, 764-768.

[11] Raman, K. (2000). The Laplace Rule of Succession Under a General Prior, Interstat, 4-10.

[12] Rusu, O., Halcu, I., Grigoriu, O., Neculoiu, G., Sandulescu, V., Marinescu, M. and Marinescu, V. (2013). Converting unstructured and semi-structured data into knowledge, Proceedings - RoEduNet IEEE International Conference, Romania.

[13] Tirunillai, S. and Tellis, G. (2012). Does chatter really matter? dynamics of user-generated content and stock performance, Marketing Science, 31, 198–215.