IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 7 Issue 7, July 2020
ISSN (Online) 2348 – 7968 | Impact Factor (2020) – 6.72
www.ijiset.com

# An Analysis of Classification Algorithms for Nepali News

**Kamal Acharya[1], Subarna Shakya[2]**

[1]Science and Technology Department, Purbanchal University
[2]Department of Electronics and Computer Engineering, Tribhuvan University

***Abstract*— This study compared different classification algorithms namely SVM-RBF Kernel, SVM-Poly Kernel, NB Multinomial and Random Forest. Datasets were prepared using web crawler from various Nepali news portals as well as from online repository on kaggle.com. News classification task began with news collection. After that pre-processing was done using Natural Language Tool Kit (NLTK) in which special symbols and stop words were removed, tokenization of keywords was done. Word stemming was carried out with the help of Lovins Stemmer. Finally four different classification algorithms (SVM-RBF Kernel, SVM-Poly Kernel, NB Multinomial and Random Forest) were implemented and compared on the basis of evaluation metrics Accuracy, Precision, Recall and F-Measure. Among them SVM-Poly Kernel outperformed remaining three algorithms with Accuracy 82.76%, Precision 82.9%, Recall 82.8 % and F-Measure 82.7%.**

***Keywords*— Nepali News Classification, SVM-Poly Kernel, SVM-RBF Kernel, NB Multinomial, Random Forest**

## I. Introduction

In our daily life there is lots of data in different field. Whenever there is data we can have lots of information, patterns, meaning etc. The information can be stored in computer in the form file, database or data warehouse. Moreover, this information helps us to extract knowledge for decision making. Good decision making process helps us identifying, selecting, and implementing alternatives. The right information, in the right form, at the right time is needed to make good decisions. The process of Extracting or “mining” knowledge from large amount of data is called Data mining [1]. Data mining also can be defined as Exploration and analysis of large quantities of data to discover meaningful pattern from data and is also known as “Knowledge discovery from data (KDD)” [1].

In data mining [1] there are lots of techniques to mine the knowledge from data which are recently used widely in different fields such as Business, Scientific Research, Computer Science, Machine Learning, Information Science, Statistics, and Database Technology etc. Most commonly used data mining techniques are Classification, Regression, Clustering and Dependencies and Associations.

Online news portal and other media on the internet now produced the large amount of text, which is mostly unstructured in nature. When an individual wants to access or share particular news, it should be organized or classified in the proper class. Automatic classification of text is to assign a label or class to given text using a computer program [2].

Data mining applications has got rich focus due to its significance of classification algorithms. The comparison of classification algorithm is a complex and it is an open problem. First, the notion of the performance can be defined in many ways: accuracy, speed, cost, reliability, etc. Second, an appropriate tool is necessary to quantify this performance. Third, a consistent method must be selected to compare with the measured values.

At present, as like in all other parts of the world, the most of the news now flashed out from the online media in Nepal. The online news portals classify their news into different categories such as "Political News", "Sports News", “Entertainment News" and so on. This task of manually labelling the news class becomes tedious when a large amount of news comes together from heterogeneous sources. It is almost impossible to make this classification manually if some application tries to feed the trending news to the reader in real time [2]. Hence the selection of the best classification algorithm for the development of an automatic tool that will be able to classify the Nepali news into relevant class is a measure problem.

This research compared the different classification algorithms (Random Forest, Naïve Bayes Multinomial, SVM-RBF Kernel and SVM-Poly Kernel) for classifying Nepali news so that the best algorithm can be implemented in the automatic tool.

## II. Related Works

In research work [2] author had evaluated some most widely used machine learning techniques, mainly Naive Bayes, SVM and Neural Networks, for automatic Nepali news classification problem. To experiment the system, author used a self-created Nepali News Corpus with 20 different categories and total 4964 documents, collected by crawling different online national news portals. TF-IDF based features were extracted from the pre-processed documents to train and test the models. The average empirical results showed that the SVM with RBF kernel was outperforming the other three algorithms with the classification accuracy of 74.65%. Then followed the linear SVM with accuracy 74.62%, Multilayer Perceptron Neural Networks with accuracy 72.99% and the Naive Bayes with accuracy 68.31%.

In [14] the researcher intended to find the appropriate algorithm to automatically classify a news articles in Indonesia Language. They compared the TF-IDF and SVD algorithm for feature selection, while also compared the Multinomial Naïve Bayes, Multivariate Bernoulli Naïve Bayes, and Support Vector Machine for the Classifiers. Based on the test results, the combination of TF-IDF and Multinomial Naïve Bayes Classifier gave the highest result compared to the other algorithms, with precision 0.9841519 and recall 0.9840000.

In [15] task of classifying documents into predefined categories was carried out. This paper compared different text classification methods based on their effectiveness on the Nepali language. Results from 3 models, SVM with word2vec and cosine similarity with TF-IDF and LSI show that the word2vec model outperforms the TF-IDF only method by 1.6 percentage and cosine similarity with LSI method by 2.2 percentage.

In [16] authors have studied the impact of text pre-processing and different term weighting schemes on Arabic text classification. In addition, developed new combinations of term weighting schemes to be applied on Arabic text for classification purposes. The stemmed and root text were obtained using two different pre-processing tools. The results illustrated that using light stemmer combined with a good performing feature selection method enhanced the performance of Arabic Text Categorization especially for small threshold values.

In research [17] researcher deled with Bangla news classification. From pre-processing the news text, they tried to do all sorts of procedures to classify the news text using Machine Learning classifier, "Naive Bayes classifier" and developed a user interface to take the news text and showed the class of that news.

## III. RESEARCH METHODOLOGY

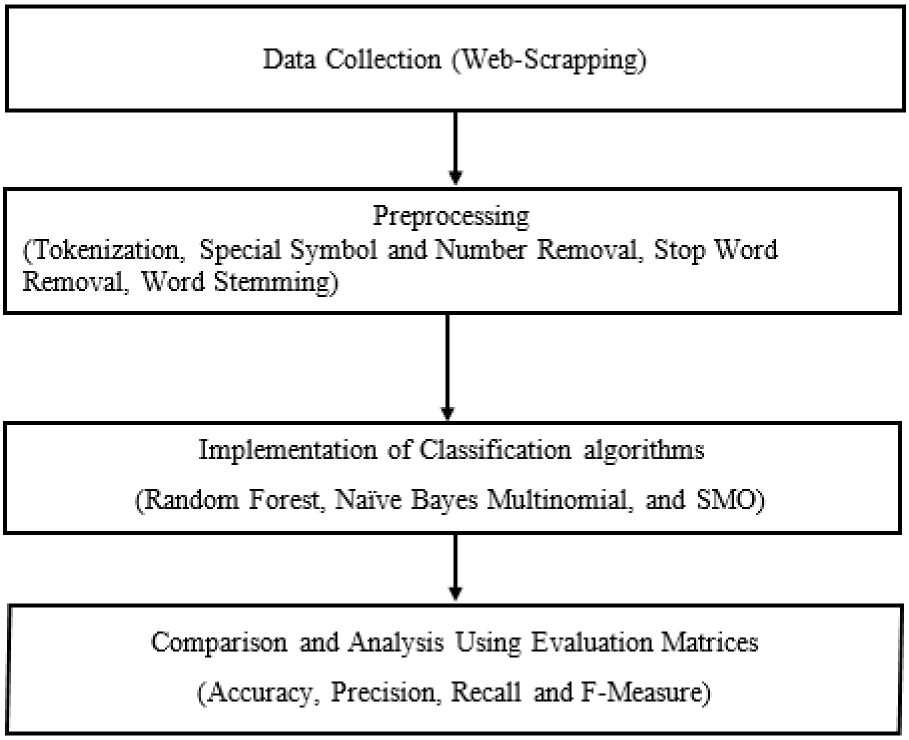

Data Collection (Web-Scrapping)
Preprocessing
(Tokenization, Special Symbol and Number Removal, Stop Word Removal, Word Stemming)
Implementation of Classification algorithms
(Random Forest, Naïve Bayes Multinomial, and SMO)
Comparison and Analysis Using Evaluation Matrices
(Accuracy, Precision, Recall and F-Measure)

Fig 1. Implementation Model

### *A. Data Collection*

Nepali news were collected from the various online news portals(onlinekhabar.com, Ratopati.com, setopati.com, Nagariknews.com, Ekantipur.com) by performing web-scrapping using the package available in the python library named BeautifulSoup. Some of the data were also obtained from the online repository (kaggle.com). Collected Nepali news corpus contained 1000 news in each 10 different classes of news.

### *B. Pre-processing*

Pre-processing was done to change the data into format that can be feed into the algorithms. For pre-processing following steps were carried out. These all the steps were carried out using the NLTK package available in python

1) *Tokenization:* Collected news were tokenize i.e separated into individual words.
2) *Special Symbol and Number Removal:* Special character like ? , ! | and the numbers like ० ,१ ,२ were removed.
3) *Stop Word Removal:* Words which don’t have special making and can be removed without altering the meaning of the sentence are stop words. In Nepali corpus stop words are छ, □□□, ल, म , □□□□ , □□□□. These were removed creating the list of the stop words for Nepali language.
4) *Word Stemming:* It is the process of obtaining the root word by removing the additional suffixes attached. For example the stemming of □□□□□ gives □□□. From among the number of available stemming algorithms LovinsStemmer was used.

### *C. Implementation of Classification Algorithms*

There are many different classification algorithms available. For this research four classification algorithms that were chosen were Random Forest, Naïve Bayes Multinomial, SVM-RBF Kernel and SVM-Poly Kernel. SVMs were implemented using SMO algorithms by changing the kernel used.

All these algorithms were implemented using WEKA (Waikato Environment for Knowledge Analysis). WEKA [12] is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from own Java code. WEKA contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes [10]. It runs in almost any platform and has been tested under Linux, Windows and Macintosh operating systems- and even on a personal digital assistant [8]. WEKA’s native data storage method is Attribute-Relation File Format (ARFF) [13]. So the data obtained after the pre-processing was changed to arff format before applying the following algorithms.

1) *Random Forest:* Random Forest [8, 9] constructs random forests by bagging ensembles of random trees. It combines more than one classifiers into one to improve the classifier’s accuracy, therefore such classifiers are called ensemble method of classifier. It combines learning method for classification and regression. It is operated by using a collection of multiple decision trees at training time and individual trees gives its own output. This algorithm was developed by Leo Breiman and Adele Cutler. It combines Breiman's "bagging" idea and Tin Kam Ho random decision forest. In this algorithm, the individual decision trees are generated using a random selection of attributes at each node to determine the split. Each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. Each tree votes and the most popular class are returned.
2) *Naïve Bayes Multinomial:* This classifier is based Bayes' theorem and computes probabilities to be able to perform Bayesian inference. The simplest Bayesian strategy, Naive Bayes, is called a special situation of algorithm that requires number adaptation to data streams. It is easy to train , and performs well when it comes to reliability and generalization, rendering it a great strategy for baseline comparison [7].

   The NB-Multinomial classifier[18] is one NB classifier variant used for multinomially distributed data like the one in the text classification. It is often used due to its easiness in implementation and execution speed.

   Multinomial Naïve Bayes or multinomial NB model, is a probabilistic learning method. The probability of a document d being in class c is computed as

   $$P(c|d) \propto P(c) \ \prod 1 \leq k \leq nd \ P(t_k|c)$$

   where $P(t_k |c)$ is the conditional probability of term tk occurring in a document of class c.[19]
3) *SVM-RBF and SVM-Poly Kernel:* Sequential Minimal Optimization (SMO) is a new algorithm for training support vector machines. Training a support vector machine requires the solution of a very large quadratic programming (QP) optimization problem. SMO breaks this large QP problem into a series of smallest possible QP problems. These small QP problems are solved analytically, which avoids using a time-consuming numerical QP optimization as an inner loop. The amount of memory required for SMO is linear in the training set size, which allows SMO to handle very large training sets. On real world sparse data sets, SMO can be more than 1000 times faster than the chunking algorithm [10].

### *D. Comparison and Analysis*

For evaluating the algorithms I have used the 5-fold cross-validation. And the confusion matrix was used for analysing the output of the algorithms.

1) *5-fold Cross-validation:* In 5-fold cross-validation, the initial data were randomly partitioned into 5 mutually exclusive subsets or “folds” i.e. D1, D2, D3, D4 and D5 each of approximately equal size. Training and testing was performed 5 times in the ratio of 4:1 means to say 4 fold as Training and 1 fold as Testing.
2) *Confusion Matrix:* A confusion matrix is a table for analyzing the result of the classifiers. It deals with how classifier can recognize tuples of different classes. In order to develop the confusion matrix, the following terms are important.

*True Positive (TP): Positive tuples that are correctively labeled by the classifier.*
*True Negative (TN): Negative tuples that are correctly labeled by the classifier.*
*False Positive (FP): Negative tuples that are incorrectly labeled as positive.*
*False Negative (FN): Positive tuples that are mislabeled as negative.*

TABLE I
CONFUSION MATRIX

| | | Predicted Class | | |
|---|---|---|---|---|
| Actual Class | | Yes | No | Total |
| | Yes | TP | FN | P |
| | No | FP | TN | N |
| | Total | P’ | N’ | P+N |

Accuracy

Accuracy of a classifiers on a given test set is the percentage of test set tuples that are correctly classified by the classifiers. It also refers to the recognition rate of the classifier that means how the classifier recognizes tuples of the various classes.

$$\text{Accuracy} = \frac{TP + TN}{P + N}$$

Precision

Precision refers to the measure of exactness that means what percentage of tuples labeled as positive are actually such.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall

Recall refers to the true positive rate that means the proportion of positive tuples that are correctly identified. It is also known as sensitivity of the classifier.

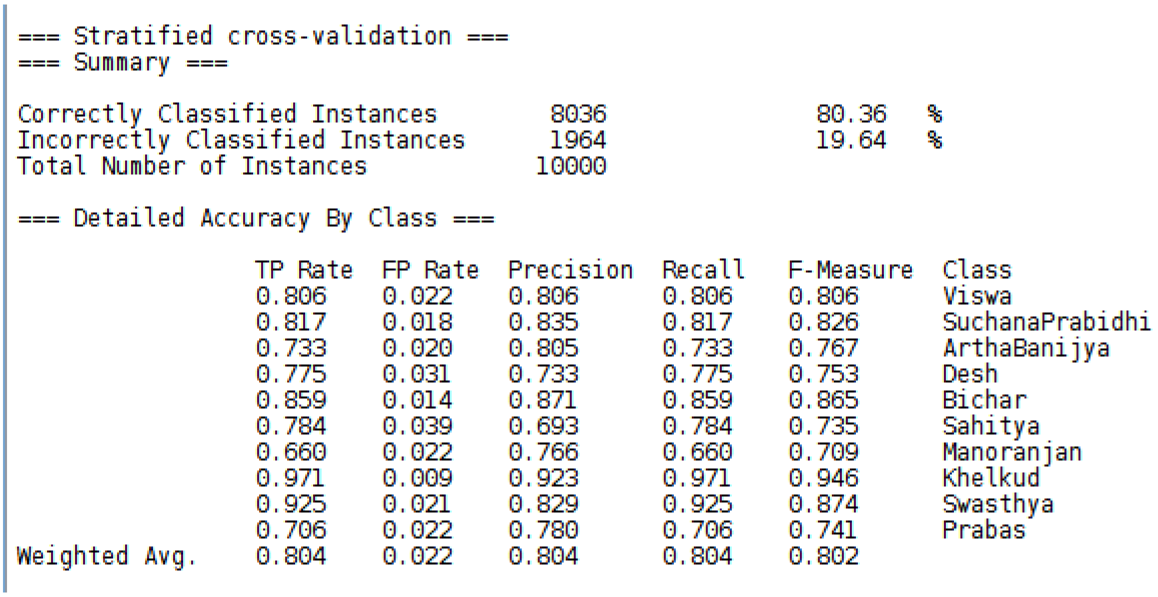
$$\text{Recall} = \frac{TP}{TP + FN} = \frac{TP}{P}$$

F-Measure

The F-score or F-Measure also refers to F-measures combines the both the measures Precision and Recall as the harmonic mean

$$\text{F} - \text{Measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precesion} + \text{Recall}}$$

The confusion matrix and the classified instances of all the four algorithms are depicted in the figures below:



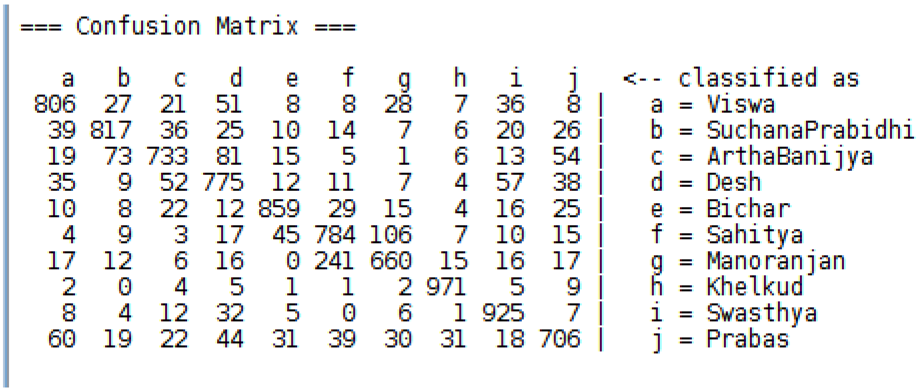
```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        8036               80.36   %
Incorrectly Classified Instances      1964               19.64   %
Total Number of Instances            10000

=== Detailed Accuracy By Class ===

                 TP Rate   FP Rate   Precision   Recall   F-Measure   Class
                 0.806     0.022     0.806       0.806    0.806       Viswa
                 0.817     0.018     0.835       0.817    0.826       SuchanaPrabidhi
                 0.733     0.020     0.805       0.733    0.767       ArthaBanijya
                 0.775     0.031     0.733       0.775    0.753       Desh
                 0.859     0.014     0.871       0.859    0.865       Bichar
                 0.784     0.039     0.693       0.784    0.735       Sahitya
                 0.660     0.022     0.766       0.660    0.709       Manoranjan
                 0.971     0.009     0.923       0.971    0.946       Khelkud
                 0.925     0.021     0.829       0.925    0.874       Swasthya
                 0.706     0.022     0.780       0.706    0.741       Prabas
Weighted Avg.    0.804     0.022     0.804       0.804    0.802
```

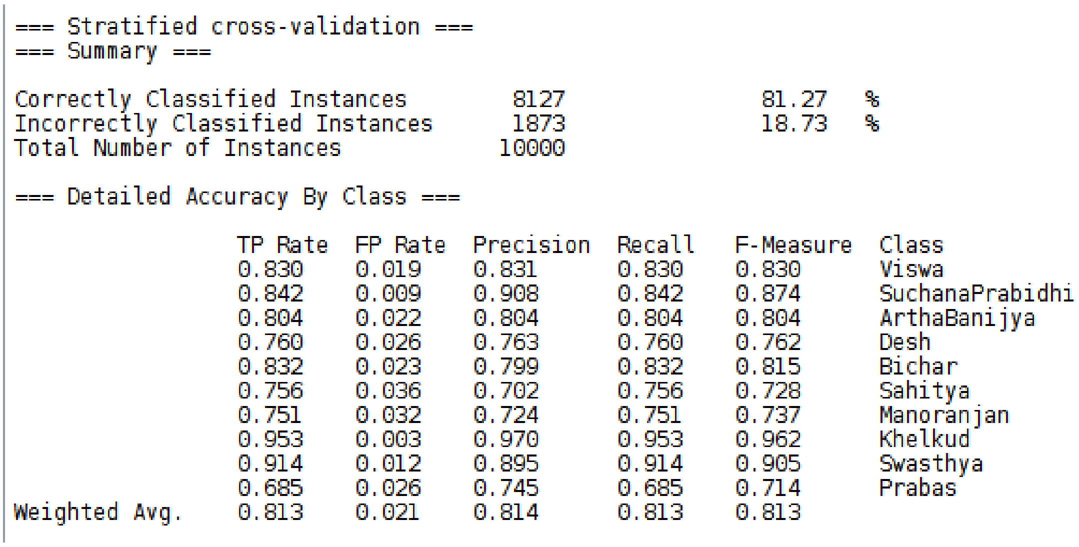
Fig 2. Classified Instances by Random Forest Algorithm



```
=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   i   j   <-- classified as
 806  27  21  51   8   8  28   7  36   8 |   a = Viswa
  39 817  36  25  10  14   7   6  20  26 |   b = SuchanaPrabidhi
  19  73 733  81  15   5   1   6  13  54 |   c = ArthaBanijya
  35   9  52 775  12  11   7   4  57  38 |   d = Desh
  10   8  22  12 859  29  15   4  16  25 |   e = Bichar
   4   9   3  17  45 784 106   7  10  15 |   f = Sahitya
  17  12   6  16   0 241 660  15  16  17 |   g = Manoranjan
   2   0   4   5   1   1   2 971   5   9 |   h = Khelkud
   8   4  12  32   5   0   6   1 925   7 |   i = Swasthya
  60  19  22  44  31  39  30  31  18 706 |   j = Prabas
```

Fig 3. Confusion Matrix of Random Forest Algorithm



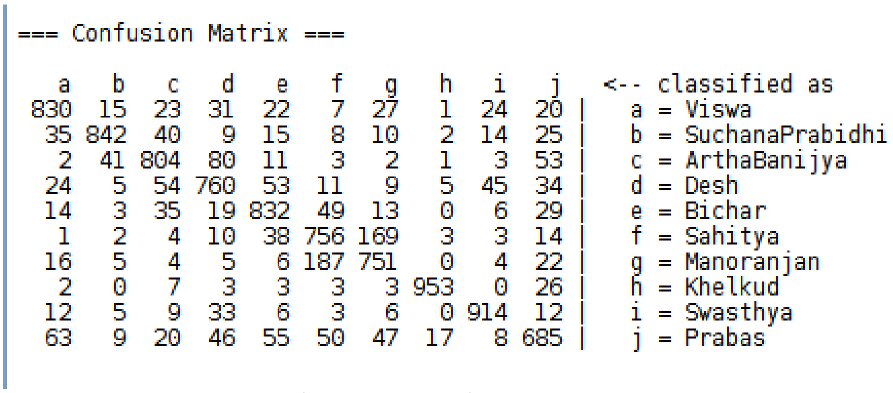
```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        8127               81.27   %
Incorrectly Classified Instances      1873               18.73   %
Total Number of Instances            10000

=== Detailed Accuracy By Class ===

                 TP Rate   FP Rate   Precision   Recall   F-Measure   Class
                 0.830     0.019     0.831       0.830    0.830       Viswa
                 0.842     0.009     0.908       0.842    0.874       SuchanaPrabidhi
                 0.804     0.022     0.804       0.804    0.804       ArthaBanijya
                 0.760     0.026     0.763       0.760    0.762       Desh
                 0.832     0.023     0.799       0.832    0.815       Bichar
                 0.756     0.036     0.702       0.756    0.728       Sahitya
                 0.751     0.032     0.724       0.751    0.737       Manoranjan
                 0.953     0.003     0.970       0.953    0.962       Khelkud
                 0.914     0.012     0.895       0.914    0.905       Swasthya
                 0.685     0.026     0.745       0.685    0.714       Prabas
Weighted Avg.    0.813     0.021     0.814       0.813    0.813
```

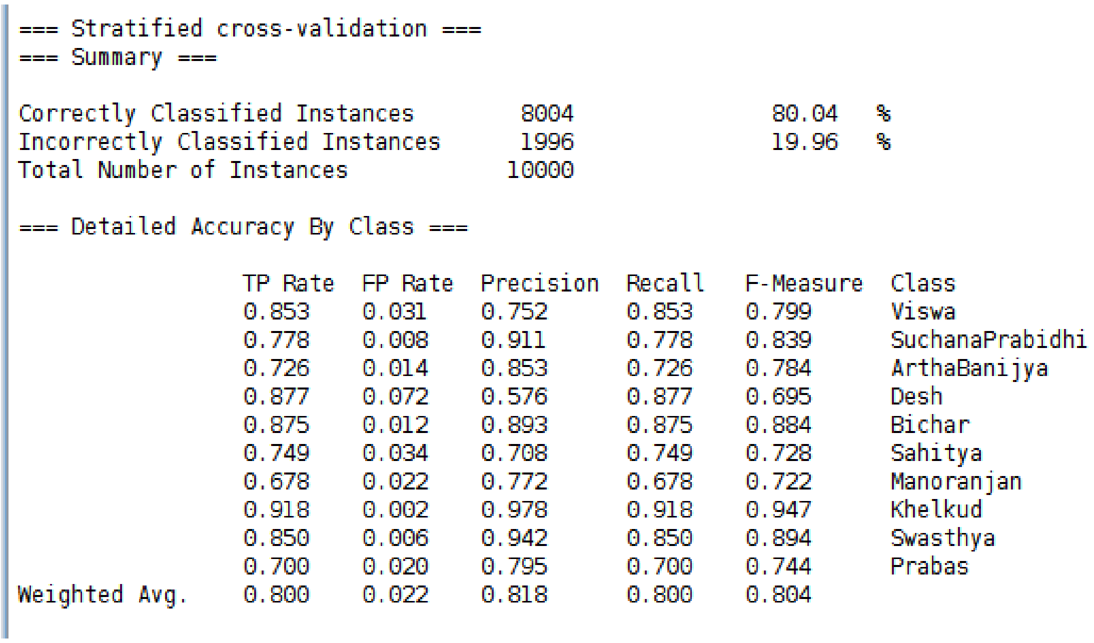
Fig 4. Classified Instances by Multinomial Naïve Bayes Algorithm



```
=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   i   j   <-- classified as
 830  15  23  31  22   7  27   1  24  20 |   a = Viswa
  35 842  40   9  15   8  10   2  14  25 |   b = SuchanaPrabidhi
   2  41 804  80  11   3   2   1   3  53 |   c = ArthaBanijya
  24   5  54 760  53  11   9   5  45  34 |   d = Desh
  14   3  35  19 832  49  13   0   6  29 |   e = Bichar
   1   2   4  10  38 756 169   3   3  14 |   f = Sahitya
  16   5   4   5   6 187 751   0   4  22 |   g = Manoranjan
   2   0   7   3   3   3   3 953   0  26 |   h = Khelkud
  12   5   9  33   6   3   6   0 914  12 |   i = Swasthya
  63   9  20  46  55  50  47  17   8 685 |   j = Prabas
```

Fig 5. Confusion Matrix of Multinomial Naïve Bayes Algorithm



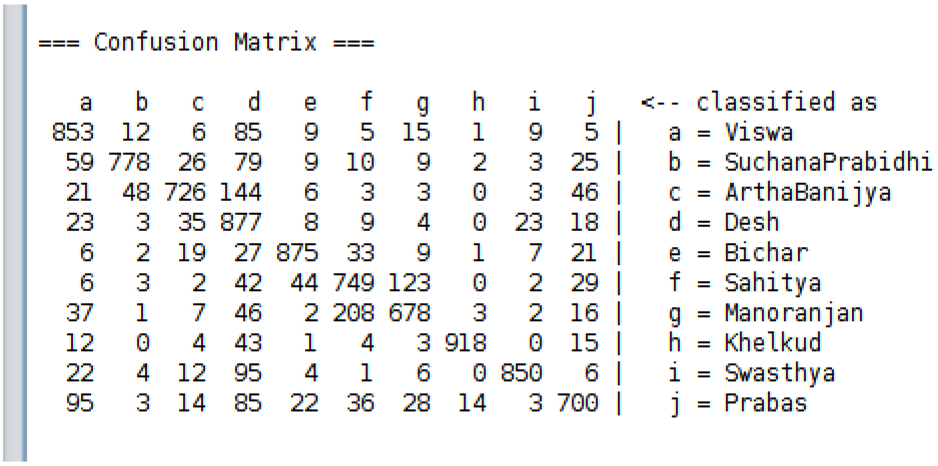
```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        8004               80.04   %
Incorrectly Classified Instances      1996               19.96   %
Total Number of Instances            10000

=== Detailed Accuracy By Class ===

                 TP Rate   FP Rate   Precision   Recall   F-Measure   Class
                 0.853     0.031     0.752       0.853    0.799       Viswa
                 0.778     0.008     0.911       0.778    0.839       SuchanaPrabidhi
                 0.726     0.014     0.853       0.726    0.784       ArthaBanijya
                 0.877     0.072     0.576       0.877    0.695       Desh
                 0.875     0.012     0.893       0.875    0.884       Bichar
                 0.749     0.034     0.708       0.749    0.728       Sahitya
                 0.678     0.022     0.772       0.678    0.722       Manoranjan
                 0.918     0.002     0.978       0.918    0.947       Khelkud
                 0.850     0.006     0.942       0.850    0.894       Swasthya
                 0.700     0.020     0.795       0.700    0.744       Prabas
Weighted Avg.    0.800     0.022     0.818       0.800    0.804
```

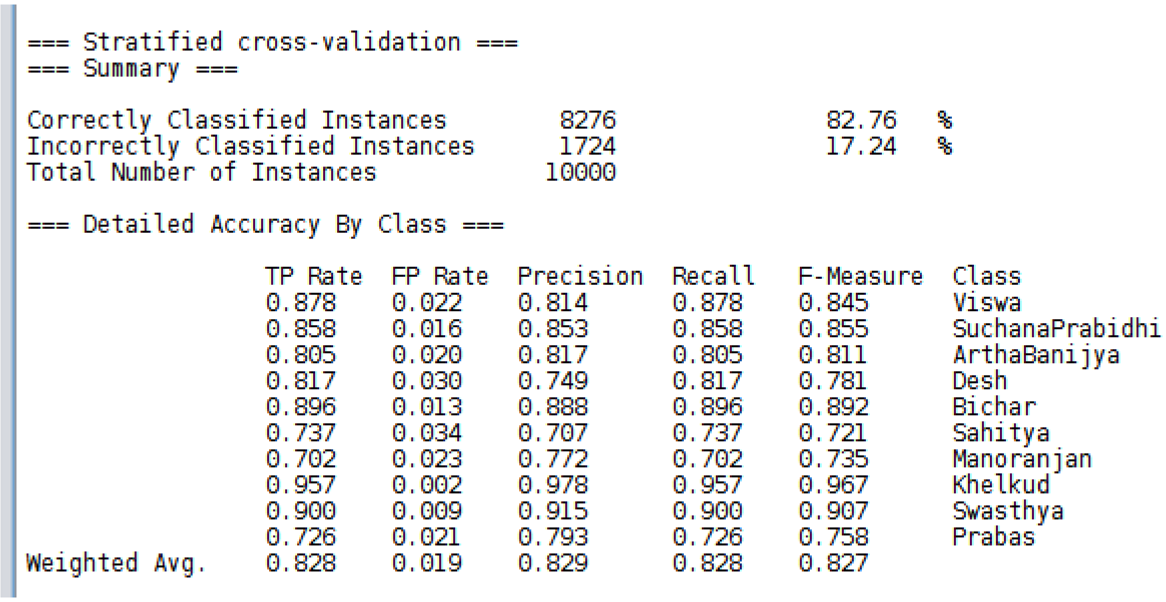
Fig 6. Classified Instances by SVM-RBF Kernel Algorithm



```
=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   i   j   <-- classified as
 853  12   6  85   9   5  15   1   9   5 |   a = Viswa
  59 778  26  79   9  10   9   2   3  25 |   b = SuchanaPrabidhi
  21  48 726 144   6   3   3   0   3  46 |   c = ArthaBanijya
  23   3  35 877   8   9   4   0  23  18 |   d = Desh
   6   2  19  27 875  33   9   1   7  21 |   e = Bichar
   6   3   2  42  44 749 123   0   2  29 |   f = Sahitya
  37   1   7  46   2 208 678   3   2  16 |   g = Manoranjan
  12   0   4  43   1   4   3 918   0  15 |   h = Khelkud
  22   4  12  95   4   1   6   0 850   6 |   i = Swasthya
  95   3  14  85  22  36  28  14   3 700 |   j = Prabas
```

Fig 7. Confusion Matrix SVM-RBF Kernel Algorithm



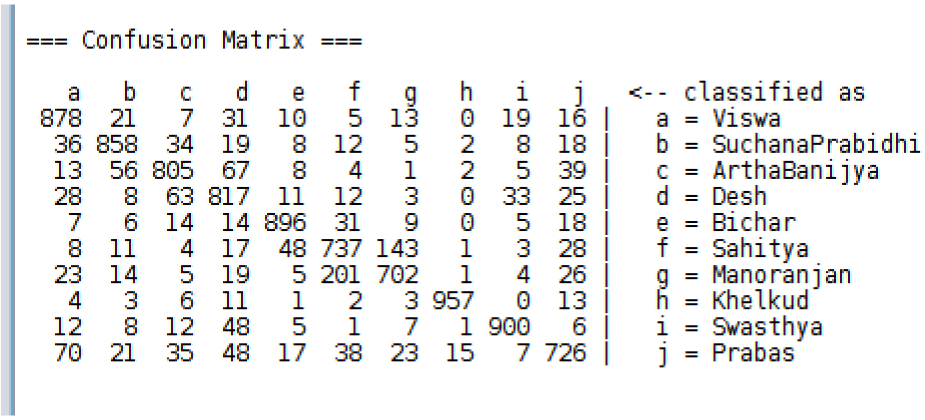
```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        8276               82.76   %
Incorrectly Classified Instances      1724               17.24   %
Total Number of Instances            10000

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  Class
                 0.878    0.022    0.814      0.878    0.845      Viswa
                 0.858    0.016    0.853      0.858    0.855      SuchanaPrabidhi
                 0.805    0.020    0.817      0.805    0.811      ArthaBanijya
                 0.817    0.030    0.749      0.817    0.781      Desh
                 0.896    0.013    0.888      0.896    0.892      Bichar
                 0.737    0.034    0.707      0.737    0.721      Sahitya
                 0.702    0.023    0.772      0.702    0.735      Manoranjan
                 0.957    0.002    0.978      0.957    0.967      Khelkud
                 0.900    0.009    0.915      0.900    0.907      Swasthya
                 0.726    0.021    0.793      0.726    0.758      Prabas
Weighted Avg.    0.828    0.019    0.829      0.828    0.827
```

Fig 8. Classified Instances by SVM-Poly Kernel Algorithm



```
=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   i   j   <-- classified as
 878  21   7  31  10   5  13   0  19  16 |   a = Viswa
  36 858  34  19   8  12   5   2   8  18 |   b = SuchanaPrabidhi
  13  56 805  67   8   4   1   2   5  39 |   c = ArthaBanijya
  28   8  63 817  11  12   3   0  33  25 |   d = Desh
   7   6  14  14 896  31   9   0   5  18 |   e = Bichar
   8  11   4  17  48 737 143   1   3  28 |   f = Sahitya
  23  14   5  19   5 201 702   1   4  26 |   g = Manoranjan
   4   3   6  11   1   2   3 957   0  13 |   h = Khelkud
  12   8  12  48   5   1   7   1 900   6 |   i = Swasthya
  70  21  35  48  17  38  23  15   7 726 |   j = Prabas
```

Fig 9. Confusion Matrix SVM-Poly Kernel Algorithm

TABLE II
COMPARISON TABLE OF THE ALGORITHMS

| S.NO | Algorithms | Accuracy (%) | Precision (%) | Recall (%) | F-Measure (%) |
|---|---|---|---|---|---|
| 1 | SVM-RBF Kernel | 80.04 | 81.8 | 80 | 80.4 |
| 2 | SVM-PolyKernel | 82.76 | 82.9 | 82.8 | 82.7 |
| 3 | NB Mutinomial | 81.27 | 81.4 | 81.3 | 81.3 |
| 4 | Random Forest | 80.36 | 80.4 | 80.4 | 80.2 |

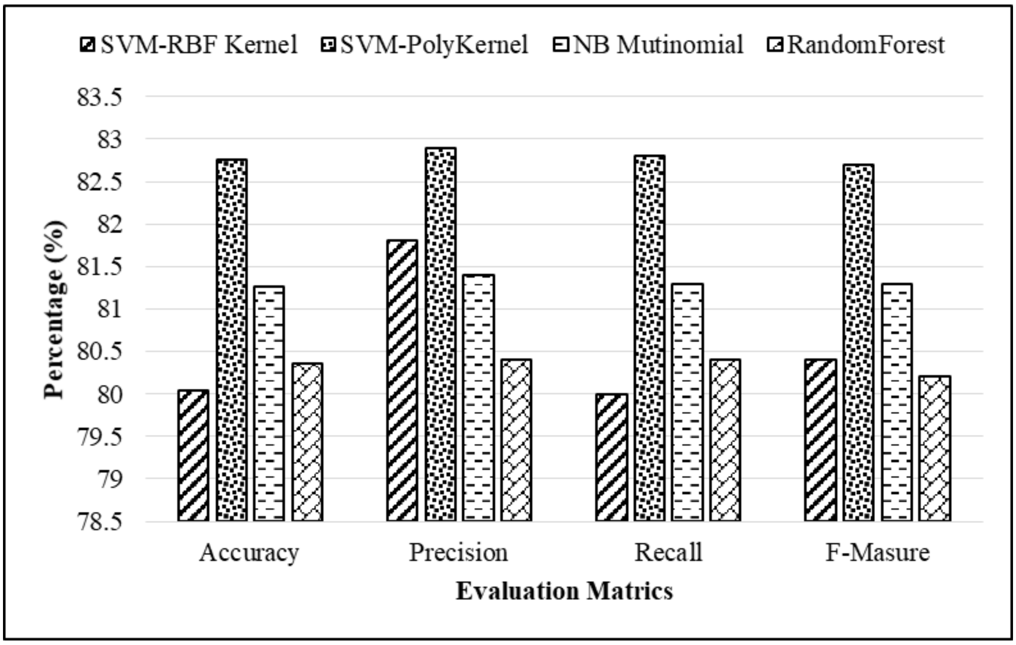


Fig 10. Graph for the table II

## IV. CONCLUSIONS

The comparison of classification algorithm is a complex task and it is an open problem. First, the notion of the performance can be defined in many ways: accuracy, speed, cost, reliability, etc. Second, an appropriate tool is necessary to quantify this performance. Third, a consistent method must be selected to compare with the measured values. The selection of the best classification algorithm for a given dataset is a very widespread problem. In this sense it requires to make several methodological choices. This research was focused in the analysis of classification algorithm for Nepali news classification where analysis was done among four classification algorithms (SVM-RBF Kernel, SVM-PolyKernel, NB Mutinomial and RandomForest).

It was found that SVM-PolyKernel was able to classify 82.76% of the data correctly which was the best among all the algorithms under comparison. In a nut shell, the result showed that SVM-PolyKernel had got about 2.72% better accuracy than SVM-RBF Kernel, 2.4% better accuracy than RandomForest and 1.49% better accuracy than NB Mutinomial for Nepali News Classification. SVM-Poly Kernel also outperformed others in term of precision, recall and F-measure.

The accuracy of the algorithms was not high enough. This accuracy can be increased by using the deep learning models and also by increasing the dataset size.

## ACKNOWLEDGMENT

It is a great pleasure for me to acknowledge the contributions of a large number of individuals to this work. I deeply extend my heartily acknowledgement to my respected teacher and worthy supervisor Prof. Dr. Subarna Shakya for his valuable guidance in carrying out this work under his effective supervision.

I would like to thank Mr. Bimal Kumar Sharma Principal HIST College and Mr. Sunil Poudel, Program Coordinator Masters’ Degree Program for their encouragement and guidance.

I would also like to express my heartfelt thanks to respected teachers, my families and friends who have helped and supported me directly and indirectly during this research.

## REFERENCES

[1] J. Han, M. Kamber, and J. Pei. "Data mining: concepts and techniques, Waltham, MA." *Morgan Kaufman Publishers* 10 (2012): 978-1.

[2] T. B. Shahi, and A. K. Pant. "Nepali news classification using Naïve Bayes, Support Vector Machines and Neural Networks." In *2018 International Conference on Communication information and Computing Technology (ICCICT)*, pp. 1-5. IEEE, 2018.

[3] "AI Horizon: Introduction to Machine Learning" Internet: http://www.aihorizon.com/essays/generalai/supervised_unsupervised_machine_learning.htm [ Sep. 20, 2018]

[4] D. Neslihan, and T. Zuhal, "A comparative framework for evaluating classification algorithms." In *Proceedings of the World Congress on Engineering*, vol. 1. 2010.

[5] J. R. Quinlan, "C4. 5: programs for machine learning." *Mach. Learn* 16, no. 3 (1993): 235-240.

[6] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan et. al., "Top 10 algorithms in data mining." *Knowledge and Information Systems* 14, no. 1 (2008): 1-37.

[7] R. Singh and D. Garg. "Hybrid Machine Learning Algorithm for Human Activity Recognition Using Decision Tree and Particle Swarm Optimization." *International Journal of Engineering Science* 8378 (2016).

[8] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal. *Data Mining: Practical machine learning tools and techniques.* Morgan Kaufmann, 2016.

[9] L. Breiman, "Random forests." *Machine learning* 45, no. 1 (2001): 5-32.

[10] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, K. R. K. Murthy. "Improvements to Platt's SMO algorithm for SVM classifier design." *Neural computation* 13, no. 3 (2001): 637-649.

[11] Microsoft Press, "Microsoft® Computer Dictionary Fifth Edition", Microsoft Press a division of Microsoft corporation, one Microsoft way, Redmond, Washington 980852-6399, ISBN: 0-7356-1495-4, May 01, 2002.

[12] "WEKA 3 - Data Mining with Open Source Machine Learning Software in Java" Internet: http://www.cs.waikato.ac.nz/ml/WEKA/ [Aug. 16, 2018].

[13] "Attribute-Relation File Format (ARFF)" Internet: http://www.cs.waikato.ac.nz/ml/WEKA/arff.html, Apr. 1, 2002 [Aug. 16, 2018]

[14] R. Wongso, F. A. Luwinda, B. C. Trisnajaya, and O. Rusli. "News Article Text Classification in Indonesian Language." *Procedia Computer Science* 116 (2017): 137-143.

[15] K. Kafle, D. Sharma, A. Subedi, and A. Kr. Timalsina. "Improving nepali document classification by neural network." In *Proceedings of IOE Graduate Conference*, pp. 317-322. 2016.

[16] M. K. Saad, and W. M. Ashour. "Arabic text classification using decision trees." *Arabic text classification using decision trees* 2 (2010).

[17] S. M. Limon, M. Ahmad, and F. N. Mishu. "Bangla News Classification Using Machine Learning." (2018).

[18] Xu, S., Zheng, W. and Li, Y. "Bayesian Multinomial Naïve Bayes Classifier to Text Classification." In: Lecture Notes in Electrical Engineering.2017

[19] C. D. Manning, P. Raghavan, and H. Schutze, "Introduction to information retrieval Cambridge University Press, 2008," Ch, vol. 20, pp. 405–416.