Proceedings of 9th IOE Graduate Conference
Peer Reviewed
ISSN: 2350-8914 (Online), 2350-8906 (Print)
Year: 2021 Month: March Volume: 9

Detecting Clickbaits on Nepali News using SVM and RF

Shiva Ram Dam [a], Sanjeeb Prasad Panday [b], Tara Bahadur Thapa [a]

[a] *Department of Information System Engineering, Gandaki College of Engineering and Science, Pokhara University, Nepal*
[b] *Department of Electronics and Computer Engineering, Pulchowk Campus, IOE, Tribhuwan University, Nepal*
Corresponding Email: [a] msc2018ise8@gces.edu.np

Abstract
Clickbaits are eye-catching headlines that are quite different from the actual content in the news. Clickbaits exaggerate the facts and lure users to click them. A dataset has been introduced that consists of Nepali news headlines and news body with label: clickbait and non-clickbait. A Machine learning model has been implemented using Support Vector Machine and Random Forest . The model uses cosine similarity metrics and TFIDF to compare between corresponding news headlines and news body, and classify them. The SVM model obtained an F1 score of 0.9483 where as RF obtained an F1 score of 0.9473. Cross validation has been used to validate the data.

Keywords
Clickbait, TFIDF, Support Vector Machine, Random Forest

1. Introduction

Digital revolution took place with the upcoming of digital media technologies like social media and smartphones [1]. The printed press has been impacted by new digital media technologies [2]. Print media is gradually transforming into digital media. This has made easy for the users to get any kind of information instantly on their hand. Readers desire to read news online through websites, blogs and social media. Online media provide a faster means of communication to people [3]. Online news sites are slowly but gradually replacing traditional form of news source like newspapers and magazines. With this shifting paradigm, online media is undoubtedly considered the future of journalism [4].

In Nepal, there are plenty of news portals and social sites [5]. They have to remain competitive in order to sustain and also to earn. One of the easiest techniques used to get more number of users is to put attractive headline that catches reader's eye quickly [6]. This generates curiosity gap in the user and forces to click on the headline. However, the content inside may be exaggerated or beyond the scope of the headline. Such are referred to as clickbaits [7].

The main purpose of clickbaits is to attract viewers to click on the catchy headlines as shown in fig. 1 This increases the number of views for that website. However, if the viewers do not find the information on the page not matching the title, their trust will be lost. Clickbait news can be found in many Nepali news portals, blogs and social sites. There are large quantity of clickbaits in Youtube Nepali videos which have catchy headlines but the content is different [8].

Figure 1: Illustration of Clickbait [9]

2. Related Works

There is extensive research done for clickbait detection. Some of the research works are performed to detect clickbait headlines solely, while some others have performed research on detection of clickbait headlines along with the contents of the news.

Pothast et al. [10] used methods of machine learning with a special emphasis on Twitter network claiming them to be the leader in this area. Logistic Regression, Naive Bayes, and Random Forest were used for their

research. Chakraborty et al. [11] used the characteristics of sentence form, n-grams, Part of Speech (POS) and special words to identify clickbaits. Using semantics and syntactics, [12] developed framework for the identification and classification of news stories as clickbait or non-clickbait. This utilized Natural Language Processing (NLP).

Cao at el. [13] developed 60 important clickbait features and performed detection of clickbait posts on social media using Random Forest (RF) classifier. Adelson et al. [14], performed clickbait detection using Parallel Neural Network (PNN). For extracting baseline features, they used Term Frequency Inverse Document Frequency (TFIDF) scores and pre-trained Gobal Vectors (GloVe) for word embedding.

3. System Model

The System model is illustrated in fig. 2. It is divided into five sections:

1. Data collection
2. Data preprocessing
3. Feature extraction
4. Algorithms
5. Model Evaluation

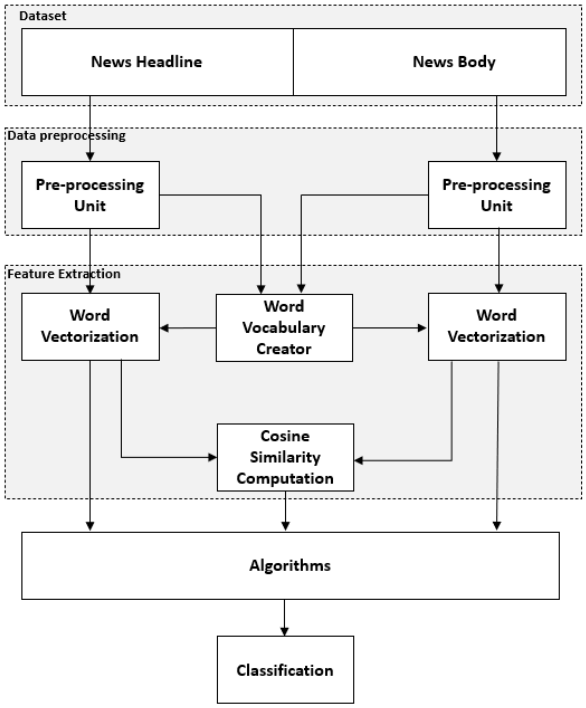

Figure 2: System Model

3.1 Data Collection

The dataset is collected from different Nepali News portals like as: KerniNews[1], DCnepal[2], GanthanNews[3], PathivaraNews[4] and social sites such as Facebook[5]. It has been prepared manually by observing and extracting clickbaits and non-clickbaits news from different news portals and social sites. 100 students from Bachelor level were requested and assigned to find at least 50 clickbaits from different Nepali news portals and social sites. The dataset consists of 10000 pairs of news pairs of title and body, out of which 4000 are labeled as "Clickbaits" and 6000 labeled as "Non-clickbaits".

3.2 Data Preprocessing

Data preprocessing was applied to both news headlines and their news body. Preprocessing of the dataset involved the phases as shown in fig. 3. Various

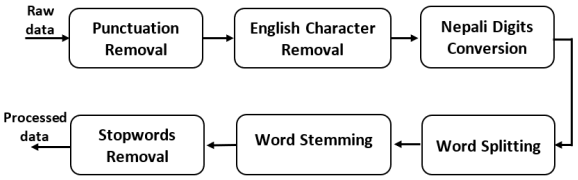

Figure 3: Data Preprocessing Process

punctuations and other special characters were removed and replaced with a blank (or white) space as shown in fig. 4. The proposed model deals with

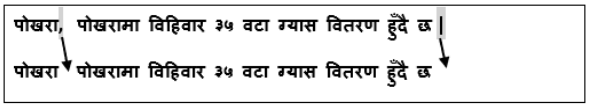

Figure 4: Punctuation Removal

Nepali language. English characters and words have very less significance. So, all the English characters and even the digits were removed and replaced by a blank space. All Nepali digits were replaced with their corresponding word form. To convert Nepali digits into its word form, a mapping technique is used which is illustrated in fig. 5. The words are split if any white space is found in the string as shown in fig. 6.

Stemming is the process of achieving the root words of inflected or derived words by eliminating suffixes and

[1]https://www.kerninews.com/
[2]https://www.dcnepal.com/
[3]https://www.ganthan.com/
[4]https://www.pathivaranews.com/
[5]https://www.facebook.com/

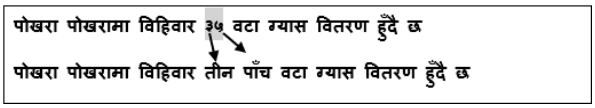

Figure 5: Nepali Digits Conversion

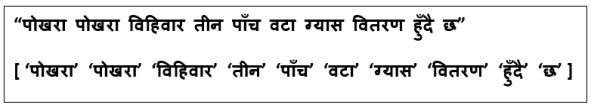

Figure 6: Word Splitting

prefixes from words in the dataset [15]. So, in order to get the main word, prefixes and suffixes had been removed as shown in fig. 7. Iteration is very important in the stemming process because a single Nepali word can have multiple prefixes and suffixes within it [15]. This stemming process is iterated for three times.

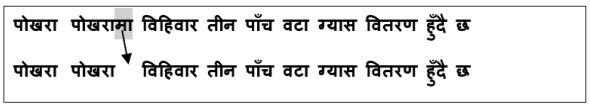

Figure 7: Nepali words stemming

Stop words are highly repeated words that puts less impact on the text [16]. Stop word is removed to boost the efficiency of the model because it hold less information. Nepali stopwords from Natural Language Toolkit (NLTK) [17] were taken. All the stopwords mentioned in the Nepali corpus from NLTK were filtered out from each corresponding news-title and news-body of the dataset as shown in fig. 8.

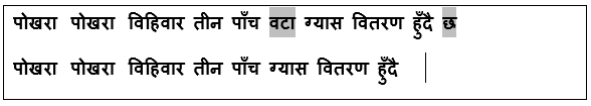

Figure 8: Stopwords Removal

3.3 Feature Extraction

3.3.1 Word Vocabulary Creation

Vocabulary here is the collection of unique words from corresponding news-title and news-body. For each pair of news-title and news-body, word vocabularies were created.

3.3.2 Word Vectorization

TFIDF have been used to represent the words in the form of vectors. The vectorization process consists of three parts:

1. Term Frequency
2. Inverse Document Frequency
3. Term Frequency Inverse Document Frequency

Term Frequency

Term Frequency (TF) is the measure of frequency of a word that appears in a document [18]. TF is given by:

$$TF(w) = \frac{N}{ND} \tag{1}$$

where, N is the count of word 'w' in a document and TN is count of total words in that document.

Inverse Document Frequency

Inverse Document Frequency (IDF) measures how important a word is. IDF is given by:

$$IDF(w) = 1 + \log\left(\frac{TND}{ND}\right) \tag{2}$$

where, TND is count of total documents and ND is count of documents that have word 'w' [18].

Term Frequency Inverse Document Frequency

TFIDF is a metric that describes the value of a word to a document compared to the whole vocabulary [18]. Mathematically:

$$TFIDF(w) = TF(w) \times IDF(w) \tag{3}$$

where, TF is term-frequency and IDF is inverse document frequency of word 'w'.

3.3.3 Cosine Similarity

Cosine similarity is a metric that calculates cosine angle between two vectors [19]. Cosine similarity calculates how similar the two documents are [20]. The cosine similarity values for different document ranges from 0 to 1. Two exactly same documents have a value of 1 and two entirely different documents have a value of 0. Other inbetween values shows intermediate similarity [20]. Cosine similarity($\cos(\theta)$) is expressed as :

$$cos(\theta) = \frac{\mathbf{A}.\mathbf{B}}{||\mathbf{A}||\,||\mathbf{B}||} \tag{4}$$

where, $\boldsymbol{A}$ is vector of TFIDF of news-title and $\boldsymbol{B}$ is vector of TFIDF of news-body.

3.4 Algorithms

The system model uses Support Vector Machine (SVM) and RF classifier for classification of clickbaits and non-clickbaits.

3.4.1 Support Vector Machine (SVM)

SVM is an useful algorithm with supervised machine learning that can be used for solving classification and regression problem [21]. Any training samples that fall on the marginal hyperplanes are the support vectors [22]. H1 and H2 in fig. 9, represent the marginal hyperplanes. A hyperplane is a line that separates and classifies into two classes: clickbait and non-clickbait. Margin is the distance between the marginal hyperplanes.

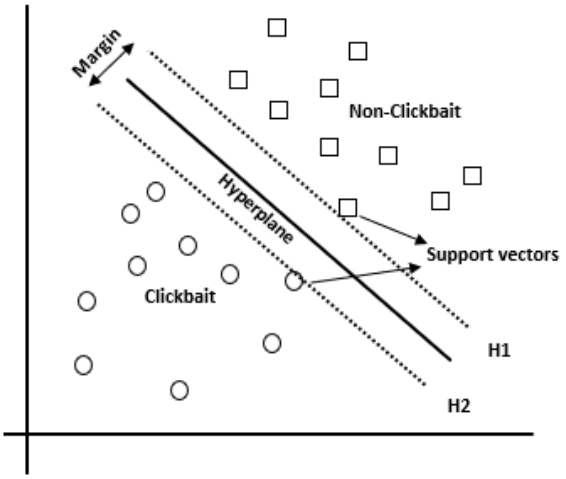

Figure 9: Support Vectors and hyperplane

The SVM takes the training dataset and finds the optimal hyperplane, and then separates all the featured data objects into two classes: Clickbait and Non-clickbait [23]. The features $\mathbf{x}_1 \ldots \mathbf{x}_n$ are the TFIDF of title, TFIDF of body and cosine similarity and the class label, $\mathbf{y}_i$ is either clickbait or non-clickbait. The output class $\mathbf{y}_i$ is classified into two classes: clickbait ($y_i = +1$) and non-clickbait ($y_i = -1$).

The Hyperplane *H* and Marginal hyperplane *H1* and *H2* equations are:

$$H : \mathbf{w}^{\mathrm{T}}\mathbf{x}_{\mathrm{i}} + b = 0$$

$$H1 : \mathbf{w}^{\mathrm{T}}\mathbf{x}_{\mathrm{i}} + b = -1$$

$$H2 : \mathbf{w}^{\mathrm{T}}\mathbf{x}_{\mathrm{i}} + b = 1$$

where, $\boldsymbol{w}^T$ represents transpose of weight vector and *b* represents bias [24].

The data points that were correctly classified should satisfy the inequality:

$$\mathrm{y_i}\ (\ \mathbf{w}^{\mathrm{T}}\mathbf{x}_{\mathrm{i}} + \mathrm{b}\ ) \geq 1 \quad \text{for} \quad \mathbf{x}_{\mathrm{i}}\ ,\ \mathrm{i} = 1, 2, \ldots .\ [24]$$

3.4.2 Random Forest (RF)

RF is an algorithm for supervised classification that generates the forest with a variety of trees for decision [25], as shown in fig. 10. The algorithm's basic concept is to construct a collection of decision trees from the randomly chosen training sets [25]. It is an ensemble tree-based learning algorithm [26]. RF algorithm constructs a large collection of decision trees, where every single tree casts a vote. The final class is decided by the majority of the votes.

Random samples from the featured dataset are taken and decision trees are constructed. Each decision tree cast a vote either to class: clickbait or class: non-clickbait. Finally, all the votes from such decisions are aggregated and final class is decided whether clickbait or not.

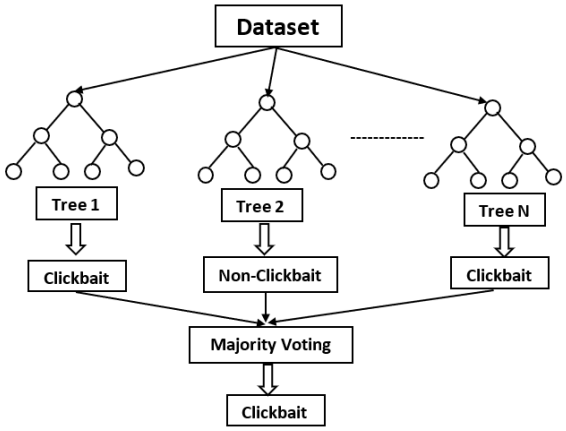

Figure 10: Structure of Random Forest Classifier

4. Model Evaluation

Train-Test split

Entire dataset with 10000 news pairs were taken. Dataset was split as Training and Testing dataset at 70:30 ratio. Model was trained using 10 fold cross validation [27] on Training dataset. Each fold consisted of 6300 Training data and 700 Validation data. The remaining Testing dataset was put for testing the model.

Cross validation

Cross-validation is a technique to uniformly distribute a dataset into train and test data repeatedly for k folds so as to prevent overfitting [28]. 10-fold cross-validation provides almost an unbiased prediction error [27]. So, a 10-fold Cross validation technique was applied for data validation.

Confusion Matrix

Confusion matrix demonstrates classifier's performance with a table of actual prediction and false predictions. [23]. A 2x2 confusion matrix was used for evaluating the performance. This matrix compares the actual target with predictions done by the model.

Evaluation metrics

Tab. 1 shows the evaluation metrics used for model evaluation.

Table 1: Evaluation Metrics

Evaluation Metrics	Formula
Precision	TP/(TP+FP)
Recall	TP/(TP+FN)
Accuracy	(TP+TN) / N
F1 Score	$\frac{2\times\text{Recall}\times\text{Precision}}{\text{Recall+Pecision}}$

Where,
TP : True Positive FP : False Positive
TN: True Negative FN: False Negative
N : No of data

5. Results

Extensive experiments were done on the Training dataset using cross-validation [27]. The models were then tested with Testing dataset.

Support Vector Machine

Fig. 11 shows performance of SVM classifier on Testing dataset. Out of 1229 clickbaits, 1130 were predicted correctly and out of 1771 non-clickbaits, 1721 were predicted correctly. The model obtained an F1 score of 0.9483 for the Testing data. The Precision, Recall, F1 score and accuracy for SVM are illustrated in tab. 2.

Table 2: SVM Model Performance

	Clickbait	NonClickbait	Average
Precision	0.9576	0.9456	0.9516
Recall	0.9194	0.9718	0.9456
F1 Score	0.9381	0.9585	0.9483
Accuracy			0.9503

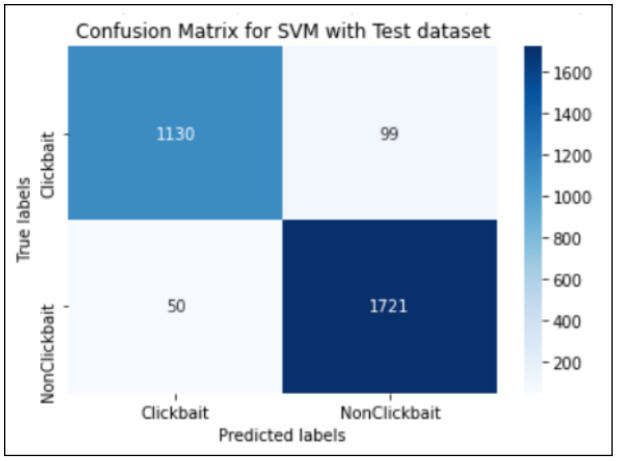

Figure 11: Confusion Matrix for SVM with Testing dataset

Random Forest

Confusion matrix shown in fig. 12 shows performance of RF classifier on Testing dataset. Out of 1229 clickbaits, 1139 were predicted correctly. Similarly, out of 1771 non-clickbaits, 1709 were predicted correctly. An F1 score for the Testing dataset was obtained as 0.9473. The Precision, Recall and F1 score and accuracy in tab. 3, shows the summary of result.

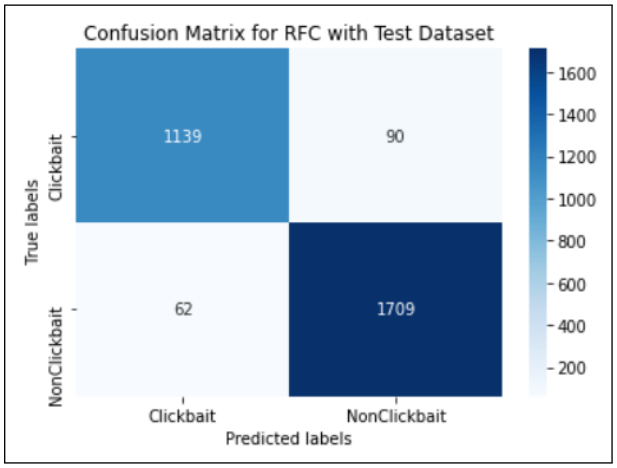

Figure 12: Confusion Matrix for RF with Testing dataset

Table 3: RF Classifier Performance

	Clickbait	NonClickbait	Average
Precision	0.9484	0.9499	0.9492
Recall	0.9268	0.9650	0.9459
F1 Score	0.9374	0.9574	0.9473
Accuracy			0.9493

Comparative Analysis

Fig. 13 shows the comparison between SVM and RF classifier in terms of accuracy, precision, recall and F1 score with Testing dataset. Both the models show nearly comparable performance. However, SVM performed slight better than RF classifier. Compared to [13] that showed F1 score of 0.61 in clickbait detection using TFIDF and [14] showed 0.65 F1 score, our model achieved better result.

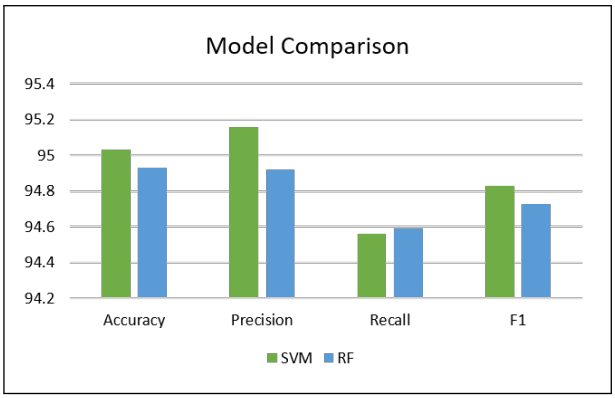

Figure 13: Model Comparison

Conclusion and Future Scope

The trend of clickbait is increasing in online Nepali media. To address this problem, a 10K dataset has been prepared. The result shows that the proposed model detects clickbait or non-clickbait with 95.03 % accuracy in Nepali news.

Future scope includes gathering more data and getting better accuracy with better model. Morphological and pragmatic analysis in the news-title and news-body of the dataset can be done before representing them into vector form. This research can provide a base for clickbait detection in Nepali Youtube videos.

Acknowledgments

The authors are grateful to Gandaki College of Engineering and Science for the provision of research facilities.

References

[1] Saumya Pandey and Gagandeep Kaur. Curious to click it?-identifying clickbait using deep learning and evolutionary algorithm. In *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1481–1487. IEEE, 2018.

[2] William Lesitaokana and Eno Akpabio. Traditional versus online newspapers: The perspective of news audiences in botswana. *Journal of applied journalism & media studies*, 3(2):209–224, 2014.

[3] Nurrida Aini Zuhroh and Nur Aini Rakhmawati. Clickbait detection: A literature review of the methods used. *Register: Jurnal Ilmiah Teknologi Sistem Informasi*, 6(1):1–10, 2020.

[4] Ujjwal Acharya. Online media in nepal: Need for policy intervention. *Policy*, 2012, 2012.

[5] Sanjeev Dulal. Top 5 online news portals of nepal. `https://www.nepalitelecom.com/2018/01/top-5-online-news-portal-nepal.html`, (Dec 2020)[Online].

[6] Amol Agrawal. Clickbait detection using deep learning. In *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, pages 268–272. IEEE, 2016.

[7] C. Hoffman. What is clickbiat? (check all that apply). In *Stupid humanism: Folly as competence in early modern and twenty-first-century culture*, pages 109–128, 2017.

[8] Aditi Aryal. Is nepali media failing to abide by ethical journalistic standards in its quest for clicks and breaking news? `https://kathmandupost.com/national/2020/05/20/`, (Dec 2020)[Online].

[9] Kerni news. `https://www.kerninews.com/category/news/page/216/`, (Dec 2020)[Online].

[10] Martin Potthast, Sebastian Köpsezl, Benno Stein, and Matthias Hagen. Clickbait detection. In *European Conference on Information Retrieval*, pages 810–817. Springer, 2016.

[11] Abhijnan Chakraborty, Bhargavi Paranjape, Sourya Kakarla, and Niloy Ganguly. Stop clickbait: Detecting and preventing clickbaits in online news media. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 9–16. IEEE, 2016.

[12] Suraj Manjesh, Tushar Kanakagiri, P Vaishak, Vivek Chettiar, and G Shobha. Clickbait pattern detection and classification of news headlines using natural language processing. In *2017 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)*, pages 1–5. IEEE, 2017.

[13] Xinyue Cao, Thai Le, et al. Machine learning based detection of clickbait posts in social media. *arXiv preprint arXiv:1710.01977*, 2017.

[14] Peter Adelson, Sho Arora, and Jeff Hara. Clickbait; didn't read: Clickbait detection using parallel neural networks. 2018.

[15] Bal Krishna Bal and Prajol Shrestha. A morphological analyzer and a stemmer for nepali. *PAN Localization, Working Papers*, 2007:324–31, 2004.

[16] Tej Bahadur Shahi and Ashok Kumar Pant. Nepali news classification using naïve bayes, support vector

machines and neural networks. In *2018 International Conference on Communication Information and Computing Technology (ICCICT)*, pages 1–5. IEEE, 2018.

[17] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[18] Gobinda G Chowdhury. *Introduction to modern information retrieval*. Facet publishing, 2010.

[19] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge, 2008.

[20] Amit Singhal et al. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, 24(4):35–43, 2001.

[21] Sujan Tamrakar, Bal Krishna Bal, and Rajendra Bahadur Thapa. Aspect based sentiment analysis of nepali text using support vector machine and naive bayes. *Technical Journal*, 2(1):22–29, 2020.

[22] 1.4. support vector machines — scikit-learn 0.20.2 documentation. https://scikit-learn.org/stable/modules/svm.html, (Oct 2020) [Online].

[23] Raj Bridgelall. Introduction to support vector machines. *Lecture Notes*, pages 1–18, 2017.

[24] Esperanza García-Gonzalo, Zulima Fernández-Muñiz, Paulino José García Nieto, Antonio Bernardo Sánchez, and Marta Menéndez Fernández. Hard-rock stability analysis for span design in entry-type excavations with learning classifiers. *Materials*, 9(7):531, 2016.

[25] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[26] Matthew N. Bernstein. Random forests. http://pages.cs.wisc.edu/~matthewb/pages/notes/pdf/ensembles/RandomForests.pdf, (Mar 2017) [Online].

[27] Annette M Molinaro, Richard Simon, and Ruth M Pfeiffer. Prediction error estimation: a comparison of resampling methods. *Bioinformatics*, 21(15):3301–3307, 2005.

[28] Daniel Berrar. Cross-validation. *Encyclopedia of bioinformatics and computational biology*, 1:542–545, 2019.