Tribhuvan University

Institute of Science and Technology

SENTIMENT ANALYSIS ON ELECTION-BASED NEPALI TWEETS

Dissertation

Submitted By

Durga Pokharel

T.U. Regd.No.:5-2-0049-0109-2015

Under the Supervision of

Associate Professor Dr. Bal Krishna Bal

Submitted to

School Of Mathematical Science

Balkhu, Kathmandu, Nepal

In partial fulfillment of the requirements for the degree of

Master in Data Science

1 March, 2024

Tribhuvan University

Institute of Science and Technology

School Of Mathematical Science

Student's Declaration

I hereby declare that I am the only author of this work and that no sources other than the listed here have been used in this work.

... ...

Durga Pokharel

Date: 1 March, 2024

Supervisor's Recommendation

I hereby recommend that this dissertation be prepared under my supervision by **Ms. Durga Pokharel** entitled **"SENTIMENT ANALYSIS ON ELECTION-BASED NEPALI TWEETS"** in partial fulfillment of the requirements for the Master in Data Science be processed for the evaluation.

... ...

Dr. Bal Krishna Bal

Associate Professor and Head of Department

Department of Computer Science and Engineering

Kathmandu University

Date: 2024/3/2

Tribhuvan University

Institute of Science and Technology

School Of Mathematical Science

LETTER OF APPROVAL

We certify that we have read this dissertation and in our opinion it is satisfactory in the scope and quality as a dissertation in the partial fulfillment for the requirement of a Master's Degree in Data Science.

... ...
Asst. Prof. Nawaraj Paudel
(Director)
School of Mathematical Science
Tribhuvan University, Nepal

... ...
Assoc. Prof. Dr. Bal Krishna Bal
Associate Professor & Head of Department
Department of Computer Science and Engineering,
Kathmandu University
(Supervisor)

... ...

(External Examiner)

... ...

(Internal Examiner)

Date.....

ACKNOWLEDGEMENTS

With a deep sense of gratefulness, I express my genuine thanks to my respected and worthy supervisor **Dr. Bal Krishna Bal**, Associate Professor and Head of Department & Department of Computer Science and Engineering (Kathmandu, University) for his valuable guidance in carrying out this work under his effective supervision and enlightenment.

I am also thankful to all my friends and my family for their motivation and help to complete this work.

ABSTRACT

In this research, we explored the best ML models and procedures that can perform sentiment classification for an election-based Nepali tweet. For the training of such models, a good set of data is necessary and we performed data collection using Twitter API and fixed keywords for several days before and after the elections in 2022. Furthermore, collected tweets were labeled manually in different classes. We implement a TF-IDF feature extractor which is described along with other competitive feature extractors. The research is primarily concerned with finding the best-performing Machine Learning Algorithm and eventually paving the way to perform sentiment analysis for Nepali tweets related to politics. This research is performed on a dataset containing unique 88973 samples in which we have 52689 for positive and 30229 for negative classes. The research shows experiments done to handle the imbalanced datasets and train different ML models. The strength and breakthrough in the research is on finding model Logistic Regression with the accuracy of 84.76% by doing an upsampling to handle imbalance. Furthermore, the second successive model was found to be a Random Forest with an accuracy of 81.70% with handling imbalanced data by upsampling.

Keywords:

tweet, NLP, stopwords, noise, imbalance, downsampling, upsampling, sentiment, label, regression, tree, probability

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

LIST OF ACRONYMS/ABBREVIATIONS

NLP Natural Language Processing

SVM Support Vector Machine

NB Naive Bayes

TF-IDF Term Frequency - & Inverse Document Frequency

NLTK Natural Language Toolkit

iNLTK Indic Language Toolkit

TnT Trigrams 'n' Tags

POS Part-Of-Speech

GNB Gaussian Naïve Bayes

BNB Bernoulli Naïve Bayes

XML Extensible Markup Language

MLP Multilayer Perceptron

CSV Comma-Separated Values

MNB Multinomial Naive Bayes

KNN k Nearest Neighbor

GBC Gradient Boosting Classifier

GRU Gated Recurrent Unit

SGD Stochastic Gradient Descent

ML Machine Learning

DL Deep Learning

SC Sentiment Classifier

TABSA Targeted Aspect Based Abusive Sentiment Analysi

CNN Convolutional Neural Network

SMOTE Synthetic Minority Over-Sampling Technique

API Application programming interface

IDF Inverse Document Frequency

DTC Decision Tree Classifier

RF Random Forest

TP True Positive

FP False Positive

TN True Negative

FN False Negative

CART Classification and Regression Tree

SMOTE Synthetic Minority Over Sampling Technique

SMOTE-NC SMOTE for Nominal and Continuous Features

ADASYN Adaptive Synthetic Sampling

ROSE Random Over-Sampling Examples

ENN Edited Nearest Neighbors

RBF Radial Basis Function

1 INTRODUCTION

1.1 Introduction

With the rise and availability of the internet, the significant rise in social media has been found to increase as well (DataReportal, 2023). Since social media has been a common ground for everyone to share their opinion about anything, the topic of political change and situations has also been included. The use of social media to spread the news of the election, parties, candidate, and their achievements has been taking place in recent years. Social media platforms like Twitter have emerged as a popular medium for political communication, particularly during elections and political campaigns. Political parties and candidates in Nepal have recently started using Twitter to interact with people and spread their messages. There is a need for efficient methods to assess and comprehend the mood of Nepali individuals toward political problems as political dialogue on social media platforms in that country keeps growing. Hence it is necessary and hard for a user to filter out and see the positive and neutral opinions without being biased towards topics or any parties.

The purpose of this research is to examine how well machine learning-based sentiment classification methods can reliably identify the sentiment in Nepali political tweets. The research issue is how accurately current sentiment classification methods recognize sentiment in Nepali political tweets. A mixed-methodologies research approach will be used for the proposed study, furthermore, incorporating methods for both quantitative and qualitative data analysis. First of all, we collect the dataset and then do sentiment analysis on the collected dataset.

The study started by doing a thorough evaluation of the literature on the sentimental analysis of tweets, including earlier research, methods, and tools. The review assists in the creation of a sentimental analysis model specifically designed for tweets about Nepali politics. In this study, we use different ML algorithms like Logistic Regression,

Naive Bayes, linear and non-linear types of Support Vector Machines, Decision Trees, Random forests, and different boosting algorithms. The best-performing model be judged on how well it captures the sentiment of Nepalese political tweets.

The study's ultimate goal is to shed light on how well sentimental classification methods can be created to discern the emotions expressed in Nepali political tweets. This study helped to create efficient sentimental analysis techniques that can be utilized to learn more about how Nepali residents feel about the political situations on social media. In addition to that, the findings of this study may have an impact on Nepal's political researchers, social media users, voters, political parties, candidates, and decision-makers.

1.2 Motivation

Sentimental classification is a rapidly expanding area of research in natural language processing that entails locating and extracting subjective data from text, such as views, attitudes, and feelings. To understand how Nepali social media users feel about different political topics, sentiment classification can be applied to political tweets on social media platforms. Unfortunately, there hasn't been much research on how emotive analysis may be applied to political tweets from Nepal. The motivation behind this sentiment analysis lies in the profound impact that social media sentiments can have on political landscapes. With the expansion of digital communication, Nepali people widely used platforms like Twitter to deliver their opinions, concerns, and allegiances. The wealth of information hidden in these tweets can give helpful insights into the collective mindset of the electorate.

1.3 Challenges

Analyzing the sentiment of Nepali tweets, particularly about the election tweets has some challenges. One of the major challenges is the different languages, Nepal is rich in different cultures, people speak different languages figuring out the sentiment of tweets even written in Nepali language may contain errors while writing which is a tough task. In the same way, informal talk is another challenge, Nepali people often use tweets for casual talk which may cause computer confusion to find out the emotions of people.

Furthermore, understanding the situation is another problem, people expressing their feelings using the Nepali language might depend on their culture. Words they used might not make sense without knowing their culture. Similarly jokes and funny talk, sometimes people have tweets that use jokes or words that mean the opposite of what they say this is one of the challenges for a computer. In the same way, in election time people express different opinions classifying these opinions correctly is one of the tricky tasks. Feeling according to place may change quickly these are some of the challenges while classifying tweets specially written in Nepali languages.

1.4 Problem Definition

There is a lot of research done on English text with the use of enough datasets on the topic of sentiment analysis. However, there are only a few that specialize in sentiment classification for Nepali texts and classification tasks were implemented in a lower amount of datasets (G et al., 2021). This research aims to create and provide a large amount of data In the field of Nepali Natural Language Processing Domain. The prepared dataset will be used for further research work in the field of Nepali Natural Language Processing. This research again aims to find out the best machine learning algorithms for sentiment classification on election-based Nepali tweets.

1.5 Research Questions

Research questions for this study can be demonstrated in the following points:

- What is the most effective ML algorithm to classify the Nepali Political tweets?
- What are the parameters to classify the Nepali political tweets?

1.6 Objectives

The objectives of this research can be described in two folds:

- To explore and evaluate the performance of best ML algorithms for election-based sentiment classification.
- To analyze the parameters to classify the Nepali political tweets.

1.7 Background Study

1.7.1 Natural Language Processing

Natural Language Processing is the subfield of Computer Science, that helps computers to process and understand the language we use as humans. NLP is part of artificial intelligence which enables computer processes and extracts insight for the creation of helpful responses. It basically, involves a set of steps that enables to convert text into vectors with the help of Machine Learning and Deep Learning. In NLP we tried to build interaction between humans and computers through natural language. Sentiment Analysis is one of the applications of NLP. Sentiment classification is also recognized as opinion mining. In Sentiment classification, we tried to classify the sentiment or emotional tone expressed in text in the different sentiment classes like positive, negative, and neutral sentiment. There are some approaches to performed sentiment classification some of them are,

1.7.1.1 Rule Based

As the name suggests in the rule-based approach we define various rules for obtaining the opinion, these rules are built by tokenizing the sentence of every document and testing the token, or word, for their presence. If a word is found classify positive sentiment +1 level is provided to it. Each tweet starts with a neutral score of zero and supposed as positive. Furthermore, positive scores are given if the polarity score is greater than zero similarly negative score are assigned if the final polarity score is less than zero. When we get the output of the rule-based system it will test the correctness of output. If the given input includes a word not present in the database which might help in movie review analysis, then these words are included in the database. This technique is commonly known as supervised learning where the system is trained to learn when a new input sentence is given.

1.7.1.2 Lexical Based

In a lexicon-based approach, we assume that the polarity of a sentence or document is equal to the sum of polarities of the each word or phrase. This technique relies

on pre-build lexical resources, like sentiment lexicons or dictionaries which map each word with their sentiment polarity, a scoring mechanism where we tokenize sentence or text into words, and each word is checked in sentiment lexicon. the average sentiment scores of individual words are calculated to find the sentiment score of the whole text likewise, negation handling, contextual analysis, and rule-based systems.

1.7.1.3 Machine learning

Machine learning approaches are applied by choosing the preferable algorithm and splitting the data set into training and testing sets normally in the ratio of 80 percent for the training set and 20 percent for the testing set or 70 percent for the training set and 30 percent for the testing set. Finally, the algorithm is trained with a training set with fine tuning we choose the optimized parameters and the trained algorithm is used to test the actual dataset. Some of the most done work for sentiment classification using machine learning approaches are Support Vector Machine(SVM), N-gram Sentiment Analysis, Naïve Bayes Method, Maximum Entropy Classifier, K-NN and Weighted K-NN, Multilingual Sentiment Analysis, and driven Sentiment Analysis.

1.7.2 Data Collection

The first step of any Natural Language Processing task is data collection, including gathering relevant textual information from various sources like social media, websites, news portals, and documents. The aim is to collect a contextual dataset that captures the specified pattern for a particular Natural Language Processing task. Whatever the pattern used for data collection like manually or automatically, data quality is important. Similarly, data completeness and ethical considerations are equally important.

1.7.3 Preprocessing

(Kadhim, 2018), text preprocessing is an important step in sentiment classification as well as text mining. The main target of text preprocessing is to able to change each document into vectors. Selecting keywords through the feature selection process and text preprocessing step is important for indexing documents. Uses of text preprocessing techniques is to reduce multiple forms of the word into one form. Furthermore, text

preprocessing task plays a vital role, particularly in machine learning. The fundamental preprocessing steps used in the study are,

1.7.3.1 Tokenization

The Natural Language Processing (NLP) pipeline is commonly started with the tokenization step or tokenization is the initial step. Tokenization means breaking down the document, paragraph, and sentence into smaller units also known as tokens. Tokenization is one of the most effective steps of the Natural Language Pipeline. The main focus of tokenization is saving sensitive information by replacing it with tokens or individual words. Tokenization can be performed on different levels, such as document level, sentence level, and word level. For the mentioned study, word-level tokenization is used. As the name suggests words-level tokenization involves breaking down the sentence into individual words.

Exploring the text at the word level gives the opportunity to understand the meaning of individual words. It forms the base for more complex language processing tasks.

1.7.3.2 Removal of White Space and Noise Dataset

After tokenization removal of white space and noise is another equally important step in the NLP pipeline. In raw data mainly text raw data getting rid of unnecessary data is necessary because it adds unnecessary complexity to the dataset. Deleting the white space and noise from the dataset is important because it helps to increase the performance of the model. White space commonly refers to empty spaces, tabs, or newlines in the same way noise refers to unnecessary or senseless information that degrades the accuracy of the classification or prediction model.

1.7.3.3 Stop Words Removal

Stop words are those words or lists of words in a text dataset that do not actually provide meaning when trying to classify or understand the sentiment of the dataset. In Nepali, words like stopwords and examples that on their own, do not contribute much meaning. So removal of stop words is one of the good ideas during the preprocessing step. Removal of these words makes the dataset clean and greatly impacts the

performance of the classification model. Details examples of stop words in Nepali will be discussed in the chapter 3.

1.7.3.4 Feature Extraction

After pre-processing the text, the next stage of text classification is feature extraction. The feature extraction step plays one of the most significant roles in the classification. It is the heart of the classification system as a good feature set represents the character of the class that helps distinguish it from other classes. Feature extraction aims to transform processed text data into a form that can be used as input by a machine learning system. Some of the feature extraction methods performed in the sentiment classification task are explained in the following subsections, There are three major categories of feature extraction techniques:

One Hot Encoding:

One hot encoding is the naive approach for representing words in a document collection. Suppose we have a collection of v documents with |v| dimension one hot encoding represents each word of documents using v dimension where each unique word has index 1 in this vector. When representing a unique word we set ith components of the vector to be 1 and the remaining components to be zero. It is one of the simplest ways to convert the document into vector form. However some of the challenges we faced while dealing with large datasets. It increases dimensionality in some cases which may cause memory issues as well as the model will take a long time to train.

TF-IDF:

Term Frequency-Inverse Document Frequency is a numerical method of testing the importance of terms within a document (corpus). It is commonly used in Natural Language Processing and information retrieval to determine the importance of words or terms in a collection of words. They are used when we need to extract the core words from documents, find ranking, etc. TF-IDF is composed of two terms, TF and IDF.

TF is the relative frequency of a term t within a document d. In other words, we can say TF as the result of the occurrence count of term t in the document d divided by the total number of words in the document d. The higher the TF value for a term t,

the higher it is repeated in the document d. Mathematically, TF can be written as follows:

$$TF = \frac{n_{ij}}{\sum_k n_{kj}} \tag{1.7.1}$$

Where n_{ij} indicates the number of occurrences of the word t_i in the document d_j similarly, $\sum_k n_{kj}$ indicate the total number of occurrences of the words in the document d_j. Furthermore, document frequency is calculated using the following mathematical formula,

$$DF_{ij} = \frac{\mid d_j \in D : t_j \in d_j \mid}{\mid D \mid} \tag{1.7.2}$$

Where |D| represents the total number of documents and $\mid d_j \in D : t_j \in d_j \mid$ indicate number of document that keyword t_j occurs.

IDF is the inverse of DF i.e. Document Frequency. It is a measure of the information provided by the term i.e. whether it is common or rare across all documents. It is calculated by taking the logarithm of the ratio of the total number of documents to the number of documents containing the term t. The higher the value for an IDF, the higher the rate of term t being rare across the entire corpus. For Inverse document frequency, we just take the log of the above DF which has the following equation,

$$IDF_{ij} = log(\frac{\mid d_j \in D : t_j \in d_j \mid}{\mid D \mid}) \tag{1.7.3}$$

Finally, we can calculate TF-IDF as the product of TF and IDF. Mathematically we can be written as follows:

$$TF - IDF = TF * DF \tag{1.7.4}$$

Practically, the value of TF-IDF ranges from 0 to 1. If the value is near 1, i.e. higher TF-IDF for a word or term in a document or a text, it means that the word is more frequent in this document and rare in the entire vocabulary or corpus. Hence it could be an important feature that helps to understand the content of the text. Inversely, if the value is low, the word may be common across the vocabulary or corpus or not frequent in this specific text.

Word2vec :

Word2vec is another way of embedding words using a neural network. Word embedding is the most popular representation of document vocabulary. Word embedding is capable of capturing the context of the word in a specific document, and semantic and syntactic similarity relation with another word in a document. Word embedding is a vector representation of a word in a document. Word2vec framework learns word associations from word corpus so that the model is capable of finding similar words or giving hints for extra terms for incomplete text after being trained. Mathematical functions like cosine similarity can be used to determine semantic similarity within words in a document with word2vec.

1.7.4 Various Methods To Handle Data Imbalance

Having an uneven number of samples for individual classes in data is common and here are the most popular ways to handle an imbalanced dataset.

1.7.4.1 Upsampling

It is one of the techniques applied to handle imbalanced text datasets, in the case of one class having comparatively fewer samples than another class. This causes biased different classifiers that work poorly on minority classes. With upsampling, we can increase the number of samples of under-represented data by randomly duplicating the number of samples required to make all classes equal in size. There are common upsampling methods like Random Oversampling, SMOTE, SMOTE-NC, ADASYN, ROSE, and Randomized SMOTE.

1.7.4.2 Downsampling

The next approach to handle the imbalanced text datasets is downsampling. Apart from upsampling which increases the minority class by downsampling, we can decrease the number of samples of over-represented data by randomly purging the number of samples required to make all classes equal in size. There are some used downsampling techniques like Random Undersampling, Cluster-Based Undersampling, Tomok Links, ENN, and Nearmiss.

1.8 Contribution of this Thesis

The main contribution of this thesis to the field of Nepali Sentiment classification in Natural Language Processing (NLP) can be seen in its extensive experimental work. A more detailed list of the various contributions is provided below,

- Use of Different Machine Learning Algorithms to Classify Sentiment of Nepali Texts.
- Investigation of feature extraction techniques for Sentiment Analysis of Nepali text.
- Identification of best classification algorithms for this particular task.

1.9 Limitation of the Study

The research has been done for the purpose of fulfilling of the requirement for the degree of master in data science. As in sentiment analysis, the dataset for Nepali text is rarely available. We prepared the dataset for sentiment analysis, particularly for political tweets. Since the labeling has been done by a single person. There might have been biases in labeling even though there was strong supervision by the supervisor. The imbalances in the dataset might be the effect of this bias. Furthermore, while collecting the dataset only limited Nepali politics-related keywords were used. Because of that, political tweets corresponding to those keywords were taken for the study.

1.10 Outline of the Thesis

The remaining part of the document is organized as follows,

Chapter 2 describes the state of the art of sentiment classification. It includes the methods and techniques used previously in the area of sentiment classification till now.

Chapter 3 describes research methodologies used in the research as well as describes the sentiment classification system architecture. The top-level system overview along with sub-system engines are given with data flow directions. Classification algorithms used for this study are described in this part of the document.

Chapter 4 describes the experiment results of the sentiment classification systems. How we do different preprocessing, and feature extraction are described. The performance and efficiency of the proposed systems evaluated in Nepali sentiment analysis datasets are given in this section.

Chapter 5 contains the summary and future scope of the research work.

2 LITERATURE REVIEW

2.1 Previous Work

This chapter describes the research works that have been already performed on the topic of several types of sentiment analysis like aspect level, sentence level, and document level. The assessment of various research articles by several researchers guides us to perform this study and evaluate our work. Furthermore, through the review of several research articles, there is the recognition of the research gap which constructs this study.

(Tamrakar et al., 2020), performed research on the topic of Aspect Based Sentiment analysis in Nepali Texts. The objective of the study was to find the best machine learning algorithm among Support Vector Machine (SVM) and Naïve Bayes (NB) classifiers. Major sources of dataset used by authors were scrapped from Nepali websites like Hamro Patro, TechPana, and SailungOnline. Furthermore, the authors have filtered the contents among numerous categories and used only the contents from the Technology sector. In the conducted research, only 1576 sentences were used with an equal number of sentences in each positive and negative sentiment to perform experiments in a balanced positive-negative class. The labeling of the sentences was done manually and Term Frequency - Inverse Document Frequency (TF-IDF) was used to calculate the importance of the words. The preprocessing of the data was done by removing wild characters and symbols. The authors have performed a tokenization using the Natural Language Toolkit (NLTK) for the Indic Languages (iNLTK) library. In addition to that, authors have used a set of 52 stop-words to remove the stop-words and applied Trigrams 'n' Tags (TnT) POS tagger using an enhanced tagged corpus to tokenize and categorize individual words to respective POS tags. Authors have used noun words to indicate aspect and the sentiment of aspect was represented by adjective words. Authors have used 3 major algorithms, Support Vector Machines, and two

variations of Naïve Bayes (Gaussian Naïve Bayes (GNB) and Bernoulli Naïve Bayes (BNB)). Being binary the nature of classification, authors have used binary SVM and performed experiments with different linear kernels.

The evaluation of the aforementioned algorithms was performed based on the metrics: F1-Score, Recall, Precision, and Accuracy. The major findings of this research showed the BNB as the best algorithm among the 3 candidate algorithms. The best algorithm, BNB used the random state of 42 and the metrics evaluated were: f1-score as 77.7%, recall as 77.5%, precision as 78%, and accuracy as 77.5%. The second best model in this pool was found to be SVM with a regularization parameter of 0.3 and the metrics evaluated were: f1-score as 76.9%, recall as 76.9, precision as 76.9, and accuracy as 76.8%. Furthermore, the authors have also provided potential experiments that might find a better algorithm as well as performance.

The results and the approaches used by the authors have given the motivation to experiment further and find a better solution. Since the dataset used in this research is limited, there is a risk of the model not being fine-tuned. In addition to that, the experimented algorithms are in small numbers compared to the gravity of the research and the availability of the various ML algorithms. Use of the more data and different algorithms like K-Nearest Neighbor, Random Forest, Logistic Regression, and even Deep Learning could also give hints of a better model.

(Singh, 2019), researched classifying Nepali text with multiple classes. The purpose of the study was to use the latest deep learning algorithm with recent word embedding to obtain higher accuracy than that obtained from traditional machine learning algorithms. The author used a Nepali corpus prepared by the Information and Language Processing Research Lab at Kathmandu University, Nepal. The collection of documents used in the corpus was taken from books, newspapers, journals, and web text. The corpus was changed into POS using XML format. Furthermore, the author also used a method to real XML in Python for NLTK corpus reading. In addition to that, the author was able to change the corpus into word-to-vector embedding for word representation purposes. The author also used the pictorial form to show the word with the highest frequency in the training set by using Word Cloud. The dataset was prepared by scrapping the web to get Nepali news content, and files for each label were placed in respective folders. In the conducted research, 10,000 datasets were used for training purposes, and testing, only 5,000 datasets were used. The category

"National News" has a smaller dataset compared to the other categories, which is an indication of a class imbalance problem. After the removal of punctuation and symbol numbers documents labeled, which were stored in CSV format, then were vectorized using TF-IDF and work2vec.

After fine-tuning and applying L2 regularization, logistic regression with a minimum term of 1000 gave the best estimate with 77.145% accuracy. In the same way, other algorithms like SVM, Multinomial Naive Bayes (MNB), BNB, Nearest Neighbor (NN), Perceptron, Multi-Layed Perceptron (MLP) with (lbfgs/sgd/adam), Gradient Boosting Classifier (GBC), SGD Classifier (SC), and Bagging Classifier (BC) were run using default parameters. Using the GRU model, achieved 77.44% accuracy, while the perceptron model gave 78.561% accuracy.

By using the deep neural network model, the researcher has given the direction for solving the multiclass Nepali text classification problem. Since the accuracy of this research is not high, one possible improvement could be achieved by increasing the data volume. Furthermore, results showed that there was not much difference in accuracy between the traditional ML model and the deep learning (DL) model. This could be the side effect of not having enough data, overfitting of the model, poor preprocessing of data, and imbalanced class.

(Singh et al., 2020), researched aspect-based detection of abusive texts in Nepali Social Media. The main focus of the study was to assemble a data set for sentiment analysis based on aspects in the Nepali social media sector and using various ML algorithms like multilingual BERT for Aspect Term Extraction and BiLSTM for Sentiment Classification (SC). The dataset consists of comments that were taken from Nepali YouTube videos with mixed and swapped codes. The author listed ten iconic Nepali "News and Politics" YouTube channels with a greater number of subscribers. And filter out the top 10 videos released from channels. From those channels authors collected the top 100 comments with some criteria like comments must have at least one Nepali character, the comment should contain a minimum of 5 words and a maximum of 50 words, one sentence in the comments may use native (Nepali), code-mixed (Romanized), or code-switched (English + Nepali) words, emoji were removed. Collected comments were lemmatized. In the annotation process joined the word manually. Authors started training models like BiLSTM+CRF for aspect term extraction and BiLSTM and Convolution Neural Network for sentiment polarity classification. For training, testing, and

validation ratio of the dataset used was 80, 10, and 10 percent respectively. While using k fold cross validation value of k was given 5. Furthermore, training was halted if the validation loss increased after 5 iterations. Researchers had trained various categories of embedding and monolingual embedding was trained on the data.

The main discovery of the research was the creation of a new Targeted Aspect Based Abusive Sentiment Analysis dataset. The authors claimed that that dataset was the first TABSA dataset in the Nepali language. By using BERT for the Aspect Term Extraction task and BiLSTM for the sentiment classification, the task achieved 57.978 percent and 81.60 percent F1 scores respectively.

This research showed that the agreement score was large for target entity identification but less than expected for aspect term identification. Another factor contributing to the low score was the large number of grammatical and syntactical errors in the social media data.

(Adhikari et al., 2022), performed research on the topic of Nepali tweets classification in a low-resource setting related to COVID-19. The major objective of the study was to classify the tweets into eight categories and build a dashboard to visualize the respective results. Data used in the study was prepared by identifying the eight top common COVID-19-related topics on Twitter using the Nepali language. The dataset was collected automatically by setting up a method to collect Nepali tweets containing specific keywords related to COVID-19 and classify the tweets into eight topics. The dataset contains 12, 241 tweets in the Devanagari script and is manually categorized into 8 specific topics. The authors also gave inter-annotator agreement statistics on the data using four annotators to label 400 popular tweets. The methodology used for the aforementioned study was: to pre-process the tweets by using the pandas library, remove the noise dataset, remove bias from source information, eliminate the trailing space and tweets with fewer words, and at last normalize the Unicode strings. The authors also mentioned the utilization of the Indic Language multi-lingual model for preprocessing and for batch processing encoder models. Furthermore, they used a linear classifier having a dropout rate of 0.5 for training the classifier and also utilized the AdamW with 0.01 weight decay and a learning rate of $5/cross10\_5$ was used.

For the model evaluation, performance metrics like the F1 score, the area under the PR curve, and Fleiss' Kappa Score were used. While looking at the class label's

performance metrics, class "COVID Stats" had a greater f1 score of 91.3% similarly, class "Life During Pandemic" had a smaller f1 score of 0.61%. Furthermore, the result showed that MuRTL worked better than mBERT after increasing the data size from 6,952 to 12, 241. When the dataset was smaller MuRIL had 0.64 mean AUPR on the other hand mBERT had 0.65 mean AUPR. For larger datasets, MuRIL has 0.84 mean AUPR whereas mBERT has 0.81. The authors compared mBERT and MuRIL for constant normalization and the dropout rate for different training data sets and proposed the best hyper-parameters. When the training dataset was smaller MuRIL showed lower performance when the dataset increased the language family-specific models improved their performance.

The study suggests that to increase the performance of MuRTL and mBERT models need to have enough amount of data and the language family of the model gives greater benefit compared to the generic model only when we can do better fine-tuning. There will be a possibility to explore further additional language and language models for sentiment classification.

(Akhmetov et al., 2022), researched the topic Sentimental Analysis in Topic Aware of News Article. The main objective of the study was to analyze the sentimental analysis using different algorithms like Convolution Neural Network, Multi-Layer Perceptron, and Decision Tree and find the achieve the better result compared to the classical ML algorithms using topic classification model on theTengrinews.kz corpus. Using this corpus, researchers classified the Kaggle articles with two types of sentimental classification models, a topic-unaware model, and a topic-aware model. About 201,230 corpora of articles were used for the model training. The sentiment-labeled dataset was used from Kaggle. Authors applied various NLP methodologies like lemmatization, stop-word removal, vectorization, and scaling finally scaled data was used to train the different models. As data were highly unbalanced for topic balancing researchers applied Synthetic Minority Over-Sampling Technique (SMOTE).

The model was evaluated using different performance metrics like F1 score, Recall, and Precision. By using a cross-validation approach on different classic algorithms extra tree classifier gives a higher accuracy of 92% on the validation set and 0.62% on the testing set. Furthermore, the CNN model gave a 0.93% accuracy score on the testing set.

The result showed that CNN works better than classical machine learning algorithms. The major weak part of the study was the low amount of publicly available sentiment-labeled texts. Researchers also mentioned that at least 94,893 articles need to have a topic labeled corpus as suggested by Data Science. Also, the authors clearly explained the reason for the better performance of CNN over the classical models, it does not need complex feature engineering as it derives the feature from trained parameters on its own. Similarly, CNN models can scale well on the data. There is the consideration that the classical model works better in small datasets however that statement was not verified in this study.

(Shrestha and Bal, 2020), publish a research article on the topic of sentiment analysis of Nepali News media texts based on name-entity. The main focus of the study was building the route for Natural Language Processing which includes error-free Name Entity recognition and anaphora regulation. For data collection, the author scrapped news from four news portals (Kantipur Daily, NagarikNews, Online Khaber, and Setopati). The author mentions that due to the lack of availability of sentence-level sentiment annotated data for the Nepali language as well as most of the previous research based on document-level sentiment analysis, authors labeled the dataset themself manually. The author labeled a total of 3490 sentences. The author used two classes (positive, and negative) to label the dataset. Furthermore, the author found 2676 datasets for the positive class and 814 datasets for the negative class. Finally, for training author used the downsampling method to handle the imbalance class problem. In the same way, the author implements a word-to-vector framework and FastText to embed the word-to-vector primarily for featured extraction. Furthermore, the author used a corpus in 300 dimensions.

After examining author mentioned that initially, they obtained better performance through word2vec skip-gram with an 80.2% f1 score similarly skip-gram with FastText gives 78.7% performance. The author Implemented different traditional machine learning approaches to classify the sentiment like Support Vector Machine, Decision Tree, and Random Forest. Among them Support vector machine with word2vec performed well and had the performance matrix with 80.15% accuracy, 80.4% precision, 80.2% recall, and 80.2% f1 score. From this research article, we can build the idea to classify the sentiment of the Nepali text however author implements the machine learning approach into a small set of data. The author suggests that we can obtain

better performance by increasing the dataset so that we can apply the deep learning approaches too.

(Shahi and Pant, 2018), conducted the study on news classification using machine learning (Support Vector Machine, Naive Bayes) and deep learning. The main goal of the study was to classify news content into respective categories. The dataset was collected through Nepali news portals like Ratopati, Setopati, Onlinekhaber, and Ekantipur using a web crawler. The author used 20 different news categories altogether 4,964 datasets were used for the study. Different Natural Language Preprocessing steps like text tokenization, noise removal, stop word removal, and word stemming were used before going to modeling. Furthermore, for the feature vector construction term frequency-inverse document frequency (TF-IDF) was used.

With Naive Bayes author found 68.98% accuracy, 70% precision, 69% recall, and 68% f1-score. Similarly, with several experiments on Support Vector Machine(SVM), the author found a performance matrix with 75.03% accuracy, 76% precision, 75% recall, and 75% f1-score. In the same way author had done an experiment using a neural network with a parameter learning rate of 0.0025, learning rate mode is adaptive, momentum of 0.9, and hidden layer neurons is 256, and with fine-tuning author found an evaluation matrix with 73.62% accuracy, 74% precision, 74% recall, and 73% f1-score. The author mentions that for high-dimensional datasets SVM performs better than other Naive Baise and Neural Network approaches.

From this research article, we gain the motivation to do work on the Nepali text datasets. However, the author implemented the TF-IDF approach for the feature extraction approach apart from that author also used a small amount of data. If the author implemented other feature extraction methods like vectorizer, and word2vec then the result would be better.

(Sitaula et al., 2021), conduct the study on the topic, Sentiment Analysis on Nepali COVID-19-Related Tweets using Deep Learning methods. The major objective of this research study is to find out the effect of COVID-19 on personal mental health by studying sentiment expressed in tweets on platforms like Twitter especially in Nepal. The author used different machine learning models like SVM, ANN, Random Forest, Naive Bayes, Logistic Regression, and K-nearest neighbors as well as CNN models. The dataset was collected on a fixed interval of time from 2020 to 2021 (one year)

with the help of the geo-location of Nepal. Only the #COVID-19 keyword was used to extract data from Twitter and label it manually.

The author mentions that among all traditional machine learning approaches Support Vector Machine performs well. Which gives 63% precision, 54.1% recall, 54.1% f1-score, and 56.3% accuracy. Furthermore, performance access using the fastText-base CNN model was 68.1% accuracy, highest F1 score found on the negative class in the same way the lowest F1 score was obtained on the neutral class. The domain-specific CNN gave a second better performance with the highest f1-score found on the negative class.

This study gave the motivation to classify Nepali COVID-19-related tweets into multiple groups and find out the sentiment hidden in text data. However, if the author implemented the other feature extraction methods like word2vec and Glove for improved performance.

2.2 Research Gaps

After reviewing the literature discussing the research gap is crucial as it highlights the area where previous study fails and new research questions emerge. After reviewing the most common research done on the topic of sentiment analysis on Nepali texts, it was found that sentiment analysis on election-based Nepali tweets is missing, and the required datasets for the sentiment analysis for the election-based tweets were also missing. Hence, this research on the topic 'Sentiment Analysis On Election-based Nepali Tweets' is done on more amount of data and thus is expected to perform better than the previous studies.

3 RESEARCH METHODOLOGY

3.1 Data Collection

The first step of any Natural Language Processing task is data collection, including gathering relevant textual information from various sources like social media, websites, news portals, and documents. The aim is to collect a contextual dataset that captures the specified pattern for a particular Natural Language Processing task. Whatever the pattern used for data collection like manually or automatically, data quality is important. Similarly, data completeness and ethical considerations are equally important.

To do experimentation in this research work, Nepali political tweet datasets are collected. Following the acceptance of requests from the Twitter developer team, data is collected using Twitter's API. The overall procedure for the collection of datasets is described in the Algorithm 3.1.1.

Algorithm 3.1.1 Political Text Dataset Collection

1: Prepare the political keywords commonly used in Nepali social media.
2: Prepare required access to Twitter Developer API.
3: Write the necessary Python program to download tweets using the keywords.
4: Saving the tweets datasets as unique CSV files per run.
5: Combine individual CSV files and make an excel file.
6: Remove duplicates and unwanted columns then make small chunks for labeling.
7: Taking individual data chunks and labeling them manually.
8: Validate the labeled dataset by a Nepali NLP expert.
9: Datasets are ready.

3.1.1 Nepali Political Keywords Used for Data Collection

The following keywords related to Nepali politics and elections are used:

जनप्रतिनिधि, सत्तारुढ, दल, सरकार, चुनाव, नगरपालिका, देउवा, उम्मेदवारी, राजनिति, निर्वाचन, नेपाल, प्रधानमन्त्री, पार्टी, सभापति, वडाध्यक्ष, नेपाली, कांग्रेस, एमाले, भोट, मेयर, राजीनामा, बालेन्द्र, हर्क, नेतृत्व, टिकट, नेता,

प्रतिनिधिसभा, समुन्नत, अग्रगामी, चुनाव, घोषणापत्र, राष्ट्रिय, स्वतन्त्र, कमरेड, बिजय, जनता, प्रतिनिधित्व, गठ-बन्धन, गणतन्त्र, उमेदबार, क्षेत्र, बहूदलीय, प्रतिस्पर्धा, सिट, बाडफाड, षडयन्त्र, सत्ता, ज्ञापनपत्र, राजनिती, आरोप, मतदाता, प्रमुख, दल, विरोध, आयोग, सांसद, समाजवादी, प्रदेशसभा, दर्ता, भ्रष्टाचार, अभियोग, प्रमुख, विद्रोह, मंसिर ४, राप्रपा, मनोनयन, माओवादी, कम्युनिष्ट, ओली, प्रचण्ड

3.1.2 About Collected Tweets

Despite this research only being focused on finding the best model of sentiment classification, we have collected additional features of data from Twitter API as well. With the use of these additional columns, we could perform exploratory data analysis to understand some quantitative nature of data.

- **id**: Default unique id of a tweet.
- **tweet_created_at**: Date and time when the tweet was created.
- **text**: Tweet text content.
- **user**: Username of a tweet author.
- **bio**: About or Bio content of a tweet author profile.
- **location**: Provided location of a tweet author.
- **hashtags**: Hashtags used in a tweet.
- **user_mentions**: Mentions in a tweet.
- **in_reply**: Tweet id of an id where it is replied to. Only available if this tweet is a reply.
- **protected**: Is the profile of a tweet author protected?
- **followers_count**: Number of followers of a tweet author.
- **friends_count**: Number of friends of a tweet author.
- **listed_count**: Number of listed counts of a tweet author.
- **created_at**: Profile creation of a tweet author.
- **favourites_count**: Favourites count of a tweet author.

- **geo_enabled**: Is geolocation enabled by a tweet author?
- **verified**: Is a tweet author verified?
- **statuses_count**: Number of status by a tweet author.
- **coordinates**: Coordinates of a tweet author (if provided).
- **is_quote_status**: Is this tweet a quote?
- **retweet_count**: Number of retweets of this tweet.
- **retweeted**: Is this tweet a retweeted tweet?
- **lang**: Language of a tweet.
- **source**: Source device of a tweet. e.g. Android, Web, and so on.
- **place**: Place from where the tweet is made.
- **kwd**: The keyword from which this tweet was found.
- **run_date**: When was this tweet downloaded by the author?

The data collection process was frequently done from a few weeks ahead of the local-level election up to the provincial-level elections of 2022. The following figure shows the number of tweets collected on a respective day.

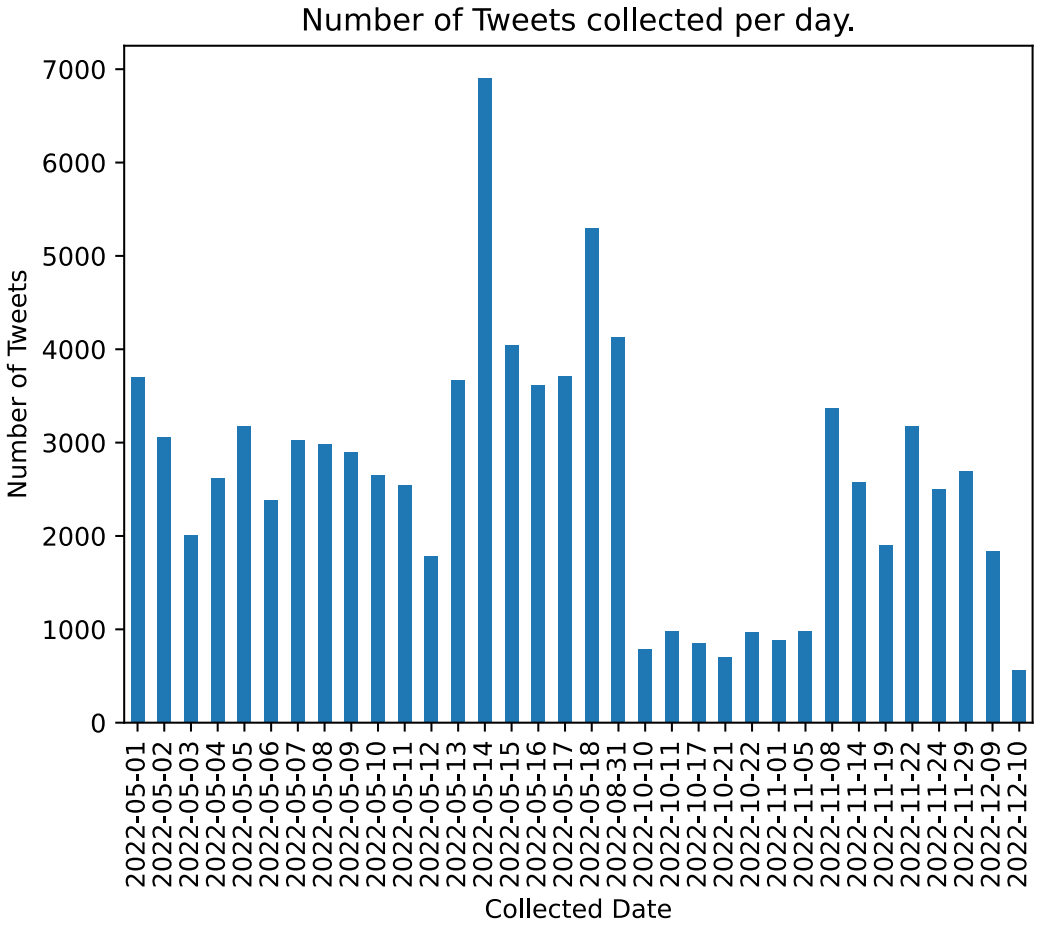

Figure 3.1: **Number of tweets collected in a respective day.**

Furthermore, the following figure shows the number of tweets created in a respective week (1 being the first week and 52 being the last week of the year 2022).

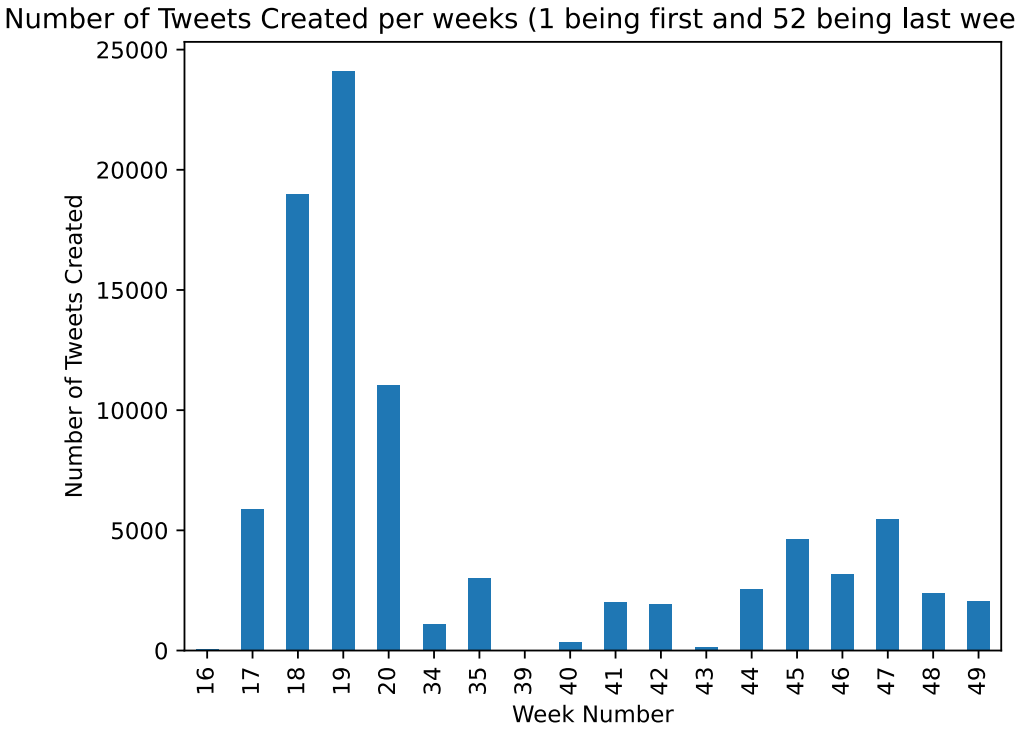

Figure 3.2: **Number of tweets Created in a n^{th} week of 2022.**

3.1.3 Sample Positive and Negative Tweets

Furthermore, tweets were labeled by the author according to the sentiment of the whole tweet sentence expresses. In addition, the labeled tweets are supervised by the supervisor of this research. Initially, four labels 0,1,2,3 for neutral sentiment, positive sentiment, negative sentiment, and out-of-political content are used. For this research work, only two labels i.e. label 1 and label 2 are used. Some of the sample positive and negative tweets collected from Twitter are given below.

नेताहरूबीच धेरै विषयमा विमति भए पनि शिखण्डी कोरल्ने 'ह्याचरी' मा भने अनवरत साझेदारी छ। यस्ता उद्योगको प्रबन्धपत्र लेखनको अग्रसरता सधैंजसो माओवादी अध्यक्ष पुष्पकमल दाहालले लिँदैआएका छन् भने शेरबहादुर देउवाको हिस्सेदारी कहिल्यै कमजोर छैन ।
https://t.co/6GYuNeHAyp

Figure 3.3: **Sample Tweet.**

नेकपा(माओवादी केन्द्र)श्रदय अध्यक्ष क.पुष्पकमल दाहाल (प्रचण्ड) संग भेटघाट प्रश्चात बधाई तथा शुभकामना व्यक्त गर्दै!!!❤

२०७९/०८/२० खुमलटार ललितपुर
@प्रचण्ड https://t.co/JSrOmBtoDW

Figure 3.4: **Sample Tweet.**

नेकपा(माओवादी केन्द्र)श्रदय अध्यक्ष क.पुष्पकमल दाहाल (प्रचण्ड) संग भेटघाट प्रश्चात बधाई तथा शुभकामना व्यक्त गर्दै!!!❤

२०७९/०८/२० खुमलटार ललितपुर
@प्रचण्ड https://t.co/JSrOmBtoDW

Figure 3.5: **Sample positive tweet 1.**

०६४ को निर्वाचनपछि पहिलोपटक नेपालको संसदमा वामपन्थी शक्तिहरू अल्पमतमा परेका छन् । कम्युनिष्ट पार्टीहरू आन्तरिक किचलोमा परेर छिन्नभिन्न भएकै कारणले प्रतिनिधिसभामा पहिलोपटक बहुमत गुमाएका हुन् ।
#nepalpress
https://t.co/pvPciAfcQi

Figure 3.6: **Sample Negative Tweet 1.**

@alive_aastha माओवादी भनेर तथानाम् भन्ने त्यस्ता बुख्याँचाहरूलाई हप्तादिन लगाएर हटट्याक हुने गरी गाली गरिदिन्छु म त । कतिपय त दोस्रो पटक ट्वीटर चलाउने आँट गर्दैनन् कि क्या हो ट्वीटरबाटै हराउँछन् । मूर्खलाई मुर्खै बनेर खेदाउनुपर्ने रहेछ।

Figure 3.7: **Sample Negative Tweet 2.**

नेकपा(माओवादी केन्द्र)श्रदय अध्यक्ष क.पुष्पकमल दाहाल (प्रचण्ड) संग भेटघाट प्रश्चात बधाई तथा शुभकामना व्यक्त गर्दै!!!❤

२०७९/०८/२० खुमलटार ललितपुर
@प्रचण्ड https://t.co/JSrOmBtoDW

Figure 3.8: **Sample Positive Tweet 2.**

3.2 System Overview

The top-level sentiment analysis system is divided into four sub-systems, text acquisition, preprocessing, feature extraction, and modeling. The top-level model of the proposed system is given in Figure 3.9.

Figure 3.9: **System Flow.**

3.2.1 Text Preprocessing

Preprocessing is done before feature extraction algorithms. The raw text is subjected to a number of preliminary processing steps to make it usable in the descriptive stages of sentiment analysis. Preprocessing aims to produce clean document text that is easy to classify sentiment accurately. The block diagram of the preprocessing system is given in Figure 3.10. Details of each preprocessing step are described in the section **??**.

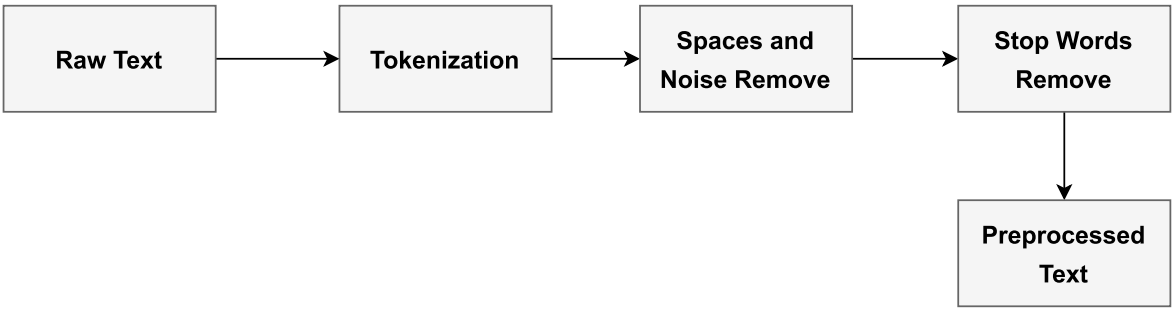

Figure 3.10: **Text Preprocessing.**

3.2.2 Feature Extraction

After preprocessing of the text data, feature vectors are extracted, which are used in the training and classification stages. Feature sets play one of the most important roles in sentiment analysis. A good feature set should represent the sentiment of a class that helps distinguish it from other classes while remaining invariant to characteristic differences within the class. The high-level block diagram of the feature extraction system is given in figure 3.11. A detailed description of each feature extraction technique is given in section 3.2.2.

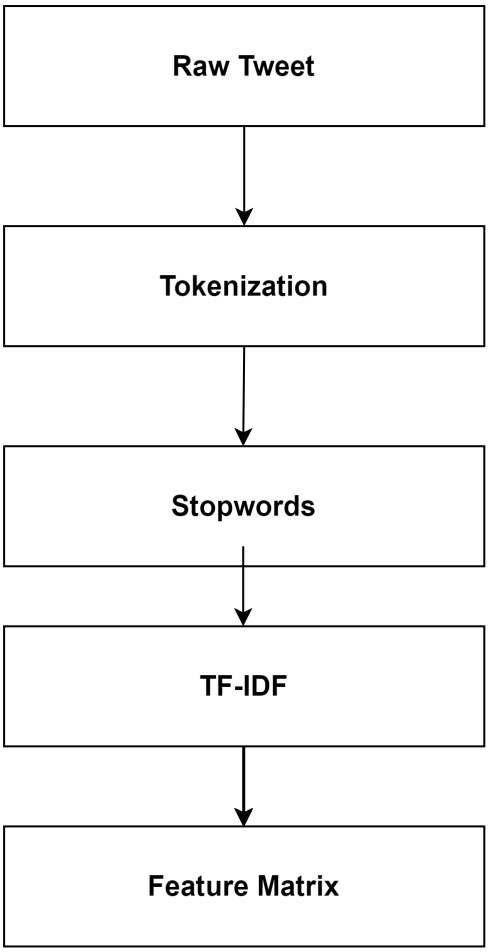

Figure 3.11: **Feature Extraction.**

3.2.3 Classification

The classification engine of the system consists of machine learning-based algorithms implemented in it. There are two stages of sentiment classification, training and testing. In the training, the stage system learns how to behave in a new environment of similar inputs and the testing stage accuracy of the classification is determined. The top-level classification system is given in figure 3.12. A detail of each algorithm used in the classification system is given in the section 3.3.

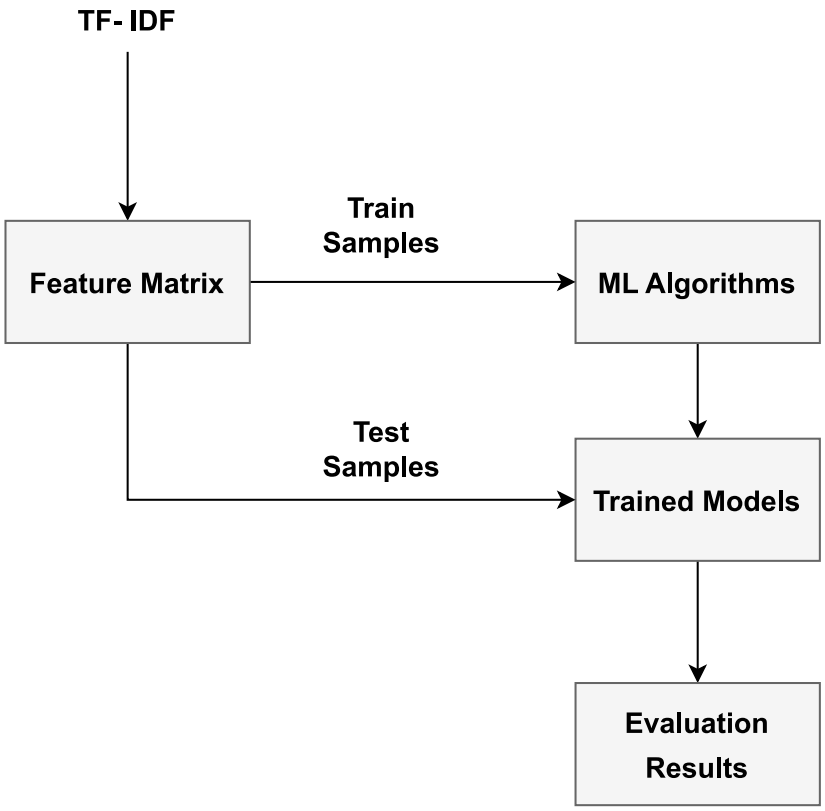

Figure 3.12: **Training-Testing System**

3.3 Training and Testing

After the feature extraction phase, the process of training and testing begins. In the training phase, the sentiment analysis system learns features of different classifiers from input feature vectors. The learning is done in a supervised manner. After the training phase, the trained model is ready to test in an unknown environment. The sentiment system is then tested against testing feature vectors and the accuracy and efficiency of the system based on classification metrics are calculated. In this research, sentiment classification is carried out using various machine learning algorithms logistic regression, Naive Bayes, Support Vector Machine, Decision Tree, Random Forest, Support Vector Machine with Radial Basis Function, K Nearest NeighborsClassifier, and various Gradient Boosting algorithms.

3.3.1 Logistic Regression

(ScienceDirect, 2024) Logistic Regression is an extension of linear regression. When the dependent variable is categorical we used logistic regression. Similarly, if the dependent variable has only two categories then we can use Binary Logistic Regression

(BLR). For sentiment classification, there are two classes positive and negative, so LR is appropriate to use. Mathematically LR is expressed as,

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_k X_k \tag{3.3.1}$$

$$P(\hat{y}) = \frac{1}{1 + e^{-y}} \tag{3.3.2}$$

Where,

- $P(\hat{y})$ is predicted value
- $\beta_0, \beta_1, \beta_2, ..., \beta_k$ are the coefficients of the logistic regression model,
- $X_1, X_2, ..., X_k$ are the variables.

Need to give a certain threshold, if we set it's value as 0.5 based on which logistic regression classifies the given data. If $\hat{y} > 0.5$ given data belongs to class 1 if $\hat{y} < 0$ given data belongs to a class 2. For logistic regression loss, the log loss function is used as an error function mathematically,

$$L(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \tag{3.3.3}$$

for the binary classification problem,

$$L(y, \hat{y}) = \begin{cases} -\log \hat{y}_i & y = 1 \\ -\log(1 - \hat{y}_i) & y = 0 \end{cases} \tag{3.3.4}$$

The weight update rule for logistic regression is given by:

$$w_i^{(t+1)} = w_i^{(t)} + \alpha \sum_{i=1}^{N} (y^{(i)} - \hat{y}) x_i$$

where,

- $w_i^{(t)}$ is the weight for the ith feature at iteration t,
- $w_i^{(t+1)}$ is the updated weight for the ith feature at iteration $t + 1$,
- α is the learning rate that controls the step size of the weight update,

- N is the number of training examples, $y^{(i)}$ is the true label of the ith training example,
- $\hat{y}$ is the predicted probability of the ith training example belonging to the positive class,
- x_i is the value of the ith feature for the training example.

Algorithm 3.3.1 Logistic Regression

1: Take input as x_i where i = 1...n.
2: Calculate the weighted input using equation (3.3.1)
3: Calculate $P(\hat{y})$ using equation (3.3.2)
4: Set threshold and classify the result based on the threshold.

3.3.2 Naive Bayes

(de Souza et al., 2022), the Naive Bayes algorithm is a machine learning model that uses the Bayes Theorem explained in equation (3.3.5) for probabilistic classification. The Naive Bayes Theorem is given by,

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{3.3.5}$$

$P(A|B)$ is called posterior probability, $P(B|A)$ is called conditional probability, $P(A)$ is prior probability, and $P(B)$ is called marginal probability. By analyzing the probability of a given set of features, known as B in the equation, the Naive Bayes classifier calculates the probability of the input data belonging to a specified class, represented as A. Also from equation (3.3.5), P(B) is a marginal probability that is constant for all of the classes, which do not contribute to the classification task hence we can ignore the denominator term $P(B)$ of equation (3.3.5).

Algorithm 3.3.2 Naive Bayes Classifier

1: Input dataset $D = (x_i, y_i)$, where $i = 1...n$.
2: Calculate the prior probability of each classes, $prior[c] = \frac{count(y_i=c)}{n}$
3: Calculate the conditional probability, $P\frac{(x_j)}{(yi=c)}$.
4: Calculate the posterior probability, $P\frac{(y_i)}{(x_j)}$. Based on posterior probability classify the data points.

3.3.3 Support Vector Machine

(Evgeniou and Pontil, 2001) Support Vector Machine (SVM) is one of the most frequently used ML algorithms which is applied to classification and regression problems as well. SVM takes input variables and gives the result as a hyperplane that separates the input variables into specified classes. For binary sentiment classification problems linear hyperplane will be utilized with the help of that hyperplane easily classify the data into two positive and negative classes respectively.

The standard linear support vector machine (SVM) can be expressed as follows:

$$\text{minimize}_{\mathbf{w},b}\frac{1}{2}\|\mathbf{w}\|^2 \tag{3.3.6}$$

subject to

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \quad \text{for } i = 1, \ldots, n \tag{3.3.7}$$

where,

- $\mathbf{w}$ represents the weight vector,
- b is the bias term,
- $\mathbf{x}_i$ denotes the i-th input data point,
- y_i represents its corresponding label (+1 or -1)

The objective is to find the hyperplane that separates the two classes with the maximum margin. This is achieved by minimizing $\|\mathbf{w}\|$ subject to the constraint that all data points are correctly classified and lie on or outside the margin boundary.

3.3.4 Decision Tree

(Wikipedia contributors, 2024) In Decision Tree Classifier (DTC) graphically, we try to classify the given input set of data points into respective classes. In that graphical tree structure, internal nodes (non-leaf nodes) denote a test on attributes, branches represent the outcomes of tests, and leaf nodes represent the class labels. There are different varieties of decision trees available for this research work we implemented

CART (Classification and Regression Trees). It works by creating a binary tree structure, and recursively dividing the datasets into subsets based on features. CART uses Gini impurity to quantify the uncertainty of datasets. The Gini impurity of a node for a decision tree is calculated using the following mathematical formulas,

$$Gini(\text{node}) = 1 - \sum_{i=1}^{C} p_i^2 \tag{3.3.8}$$

where, C is the number of classes, and p_i is proportional to the sample in class i, we try to split the dataset in a fashion that minimizes the Gini impurity. The algorithm calculates the reduction in Gini impurity. We calculate the reduction by subtracting the Gini impurity of the current nodes and the weighted sum of the Gini impurities of the child node. The maximum values of difference are chosen. Mathematically,

$$\text{Reduction} = \text{Gini(parent)} - \left(\frac{N_{\text{left}}}{N_{\text{parent}}} \cdot \text{Gini(left)} + \frac{N_{\text{right}}}{N_{\text{parent}}} \cdot \text{Gini(right)} \right) \tag{3.3.9}$$

where, n_{parent} is the number of samples in the parent node, N_{left}, and N_{right} are the numbers of samples in left and right child nodes.

3.3.5 Random Forest

Random Forest (RF) is the popular algorithm for supervised learning classes. In RF, the ensemble learning concept is applied, which is a mechanism of combining multiple decision tree classifiers to solve a problem and improve the performance of the model. These classifiers will be either a homogeneous class of model or a heterogeneous class of model. Each classifier in the RF splits out a class and the class with majority votes becomes our model's final prediction. There the concept of combining multiple learning will be utilized like, Bagging, Boosting, and Stacking.

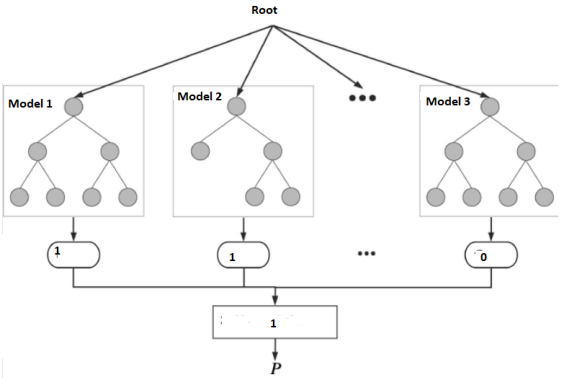

Figure 3.13: **Random Forest.**

3.3.6 Support Vector Machine with Radial Basis Function

Support Vector Machine (SVM) with Radial Basis Function (RBF) mathematical function (kernel) is the most commonly used and powerful classification algorithm in the field of machine learning. SVM are supervised class of models and can be implemented for both classification and regression tasks. RBF is a popular kernel option in SVM classifiers to handle non-linear relationships in the datasets. RBF kernel which is also known as Gaussian Kernel has the following mathematical formula,

$$K(x, x') = \exp\left(-\gamma||x - x'||^2\right) \tag{3.3.10}$$

where γ is the positive parameter that controls the width of the Gaussian. A small value of γ makes the boundary of the decision smooth similarly, a larger value of γ makes the decision boundary more complex

3.3.7 K Nearest Neighbor Classifier

The K Nearest Neighbors classifier is one of the non-parametric and lazy learning algorithms. We called it non-parametric in the sense that the model structure is determined from the dataset only. Furthermore, it does not require any training dataset for model generation. In K Nearest Neighbors classifier K is the number of nearest neighbors which is the core factor for decision. K Nearest Neighbors classifier calculates the similarity relationship between new data and training data, finds k neighbors, and keeps new data in the class that shows a similar relationship. For similarity calculation, we commonly used Minkowski distance which has the following mathematical formula,

$$D(x, y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p} \tag{3.3.11}$$

equation (3.3.11) is the combination of both Euclidean distance and Manhattan distance.

3.3.8 Gradient Boosting

Gradient Boosting is one of the well-known machine learning ensemble algorithms that sum up the strength of several weak learners in order to create a new strong predictive model. In boosting homogeneous weak learners are combined to create strong learners. Its main aim is to reduce bias which means learners who have high bias are combined to find the model with decreased bias. In boosting weak models are trained sequentially. An important characteristic of boosting is each model in the sequence is trained giving more importance to the data samples that were falsely predicted by previously used models in sequence. So that every new model has more effort on the difficult sample of datasets. Some of the examples of boosting algorithms are AdaBoost and XGBoost.

3.4 Performance Metrics

Our sentiment analysis task is the classification type of problem. Which involves classifying text on the basis of sentiment expressed in the text either in a positive or negative class. ("Confusion Matrix," n.d.), a confusion matrix is the major performance measure to evaluate how well the classification algorithms work. For binary classification task 2×2, a confusion matrix is used which is displayed in the following table. For the balanced dataset, we particularly focus on the accuracy of the model. Furthermore, for the imbalanced dataset, we focus on the f1 score. As we have an imbalanced dataset we focus on the f1-score while evaluating the performance of different ML models.

		Prediction outcome		
		p	**n**	**total**
actual value	**p′**	True Positive	False Negative	P′
	n′	False Positive	True Negative	N′
	total	P	N	

Accuracy: Accuracy is the ratio of correct predictions made by the model. It can be calculated using the following formula,

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.4.1}$$

Precision: Precision is the ratio of predicted positives that are positive and is calculated using the following formula,

$$Precision = \frac{TP}{TP + FP} \tag{3.4.2}$$

Recall: Recall is the ratio of actual positives that are correctly classified by the model and is calculated through the following formula.

$$Recall = \frac{TP}{TP + FN} \tag{3.4.3}$$

F1-Score: F1- Score is the harmonic mean of recall and precision. For the imbalance class problem, we mainly focus on the f1-Score. It is calculated using the following formula.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \tag{3.4.4}$$

3.5 Environment and Tools

All the algorithms of the proposed sentiment classification system are implemented in Python version 3.10.12. Python is used through Google Colab from Google Drive with a high RAM of storage 56 GB.

- **Python** is a high-level, interpreted, object-oriented, cross-platform, general proposed language with dynamic semantics programming language. Guido van Rossum released it on February 20, 1991.
- For the data collection we used the **Twitter Developer API**. It is a set of APIs provided by Twitter for developers to access and interact with Twitter data. **tweepy** is a Python library for accessing the Twitter API. **Google Colab** is an attractive notebook provided by Google. Here we first wrote some Python code to scrap the data from the Twitter community.
- We used **pandas** to handle and manipulate numerical tables. Scrapped tweets were stored using the pandas's dataframe and finally converted into CSV files using the Python pandas library. **NumPy** is a library in Python that provides tools for working with large arrays of numerical data and performing mathematical operations on them. The Natural Language Toolkit **(NLTK)** is a library in Python that provides tools to work with human language data (text).
- For machine learning we used **Scikit learn**(also known as sklearn). Scikit Learn is a library in Python that provides tools for machine learning and data mining tasks. Which provides tools for designing, implementing, visualizing, and simulating machine learning. In this research we used scikit learn for model evaluation as well.
- For figure drawing propose we used **Draw.io/Ms paint** to draw the figure in SVG format.

4 EXPERIMENT AND RESULTS

This research work aims to evaluate machine learning sentiment classification systems. The experiment uses different machine-learning algorithms like Logistic Regression, Naive Bayes, Support Vector Machine, Decision Tree, Random Forest, and so on. The theoretical concepts of all steps used in text preprocessing and machine learning algorithms used for this research work are described in **Chapter 3**. This chapter mainly focused on how we prepared text to transform into vectors and a comparison of the performance of different classifiers. Figure 4.1 is a sample example of raw text.

@chemjongd @rsansar @baral_bishal @Dharane111 @iSDBR @SaugatWrites
@DearGovin @sir_rosan @Bibas_H @babusasach @grpokharel @Theeigen
@AnupGaule @Janginishaan @Lakshmangurung @RabindraADH
@parajuli_aditya @g_unlucky @jibanaryal12 @Manita_Ghimire
@haridaai @iMaanava @iNepalese @ManangMaya @gkbista @gajendrajee
@gbudhathoki @Aam_Nepali @Mp_679 @thetakala @oops_u @thapagk
@SuryaRAcharya हेलिकप्टरबाटै बिउ छर्ने, ३६५ दिननै दसैँ मनाउने व्यवस्थादेखि पशु बस्तुलाई समेत अस्पतालमा सुत्केरी हुने व्यवस्था मिलाउने वाचा गरेका डडेल्धुराको भागेश्वर गाउँपालिका वडा नम्बर १ मा स्वतन्त्रतर्फ वडाध्यक्षको उम्मेदवारी दिएका खडंक ठकुराठीले ४ मत ल्याएका छन् । https://t.co/9OLyAgjKEy

Figure 4.1: **Sample raw tweet text.**

Tokenization is commonly known as the first step of the Natural Language Processing (NLP) pipeline. Tokenization is done on different levels like document labels, sentence levels, and word levels however for this study, word-level tokenization is used. Tokenization was done using white space.

```
@chemjongd', '@rsansar', '@baral_bishal', '@Dharane111', '@iSDBR',
'@SaugatWrites', '@DearGovin', '@sir_rosan', '@Bibas_H',
'@babusasach', '@grpokharel', '@Theeigen', '@AnupGaule',
'@Janginishaan', '@Lakshmangurung', '@RabindraADH',
'@parajuli_aditya', '@g_unlucky', '@jibanaryal12',
'@Manita_Ghimire', '@haridaai', '@iMaanava', '@iNepalese',
'@ManangMaya', '@gkbista', '@gajendrajee', '@gbudhathoki',
'@Aam_Nepali', '@Mp_679', '@thetakala', '@oops_u', '@thapagk',
'@SuryaRAcharya', 'हेलिकप्टरबाटै', 'बिउ', 'छर्ने,', '३६५', 'दिननै', 'दसैँ',
'मनाउने', 'व्यवस्थादेखि', 'पशु', 'बस्तुलाई', 'समेत', 'अस्पतालमा', 'सुत्केरी',
'हुने', 'व्यवस्था', 'मिलाउने', 'वाचा', 'गरेका', 'डडेल्धुराको', 'भागेश्वर',
'गाउँपालिका', 'वडा', 'नम्बर', '१', 'मा', 'स्वतन्त्रतर्फ', 'वडाध्यक्षको',
'उम्मेदवारी', 'दिएका', 'खडक', 'ठकुराठीले', '४', 'मत', 'ल्याएका', 'छन्',
'।', 'https://t.co/9OLyAgjKEy
```

Figure 4.2: **Tokenization of aforementioned tweet. Each word is represented as a token inside a single quotation and separated by a comma.**

Noise removal is one of the important steps of preprocessing in NLP. Removing white space and noise from a dataset is important because it can improve the accuracy and reliability of any subsequent analysis or modeling. White space refers to blank spaces, tabs, or newlines in the dataset. Removing white space ensures that the data is consistent and easier to work with. Similarly, noise in the dataset refers to irrelevant or meaningless data that can affect the accuracy of any classifier model. Noise data includes duplicate data, outliers, and incomplete data. In our data characters like numbers, punctuation, @, hashtags, emojis, etc also act as noise. Furthermore, we removed English words and links as well.

1. In order to remove English characters and noises like https and so on, the following regex command is used.

```
(?i)\b((?:https?://|www\d{0,3}[.]|[a-z0-9.\-]+[.][a-z]

{2,4}/)(?:[^\s()<>]+|\((?:[^\s()<>]+|(\([^\s()<>]+\)))*\))+

(?:\((?:[^\s()<>]+|(\([^\s()<>]+\)))*\)|[^\s`!()\[\]{};:'".,<>?""‘’«»]))
```

2. In order to remove the emoji we are using the following regex command.

```
[\u200c-\u200f\u202a-\u202f\u2066-\u2069]
```

3. Furthermore, the following characters are removed individually.

`%• ´=+ −÷"][{}*…‘’_&#\/; @abcdefghijklmnopqrstuvwxyz1234567890( )-
−.|!?",:?

After the completion of this step, the tweet looks like the following:

हेलिकप्टरबाटै, बिउ, दिननै, दर्सैं, मनाउने, व्यवस्थादेखि, पशु, बस्तुलाई, समेत, अस्पतालमा, सुत्केरी, हुने, व्यवस्था, मिलाउने, वाचा, गरेका, डडेल्धुराको, भागेश्वर, गाउँपालिका, वडा, नम्बर, मा, स्वतन्त्रतर्फ, वडाध्यक्षको, उम्मेदवारी, दिएका, खडक, ठकुराठीले, मत, ल्याएका, छन्

we found some words in our dataset, these words do not give meaning when they come alone so they need to be removed. These words add little or no value to the text classification. However, they only increase the memory. The most common stop words for the Nepali language are: अक्सर, अगाडि, अझै, अनुसार, अन्तर्गत, अन्य, अन्यत्र, अन्यथा, अब, अरू, अरूलाई, अर्को, अर्थात, अर्थात्, अलग, आए, आजको, आठ, आत्म, आदि, आफू, आफूलाई, आफैलाई, आफ्नै, आफ्नो, आयो, उदाहरण, उन, उनको, उनले, उप, उहाँलाई, एउटै, एक, एकदम, औं, कतै, कम से कम, कसरी, कसै, कसैले, कहाँबाट, कहिलेकाहीं, कहिल्यै, कहीं, का, कि, किन, किनभने, कुनै, कुरा, कृपया, के, केहि, केही, को, कोही, क्रमश, गए, गरि, गरी, गरेका, गरेको, गरेर, गरौं, गर्छ, गर्छु, गर्दै, गर्न, गर्नु, गर्नुपर्छ, गर्ने, गर्यौं, गैर, चाँडै, चार, चाले, चाहनुहुन्छ, चाहन्छु, चाहिए, छ, छन्, छु, छैन, छौं, छौं, जताततै, जब, जबकि, जसको, जसबाट, जसमा, जसलाई, जसले, जस्तै, जस्तो, जस्तोसुकै, जहाँ, जान, जाहिर, जुन, जे, जो, ठीक, त, तत्काल, तथा, तदनुसार, तपाइँको, तपाईं, तर, तल, तापनि, तिनी, तिनीहरू, तिनीहरूको, तिनीहरूलाई, तिनीहरूले, तिमी, तिर, ती, तीन, तुरुन्तै, तेस्रो, त्यसकारण, त्यसपछि, त्यसमा, त्यसैले, त्यहाँ, त्यो, थिए, थिएन, थिएनन्, थियो, दिए, दिनुभएको, दिनुहुन्छ, दुई, देख, देखि, देखिन्छ, देखियो, देखे, देखेको, देखेर, देख्न, दोश्रो, दोस्रो, धेरै, न, नजिकै, नत्र, नयाँ, नि, निम्ति, निम्न, निम्नानुसार, निर्दिष्ट, नै, नौ, पक्का, पक्कै, पछि, पछिल्लो, पटक, पनि, पर्छ, पर्थ्यो, पर्याप्त, पहिले, पहिलो, पहिल्यै, पाँच, पाँचौं, पूर्व, प्रति, प्रत्येक, प्लस, फेरि, बने, बन्द, बन्न, बरु, बाटो, बारे, बाहिर, बाहेक, बीच, बीचमा, भए, भएको, भन, भने, भन्ने, भन्छन्, भन्छु, भन्दा, भन्नुभयो, भन्ने, भर, भित्र, भित्री, म, मलाई, मा, मात्र, माथि, मुख्य, मेरो, यति, यथोचित, यदि, यद्यपि, यस, यसको, यसपछि, यसबाहेक, यसरी, यसो, यस्तो, यहाँ, यहाँसम्म, या, यी, यो, र, रही, रहेका, रहेको, राखे, राख्छ, राम्रो, रूप, लगभग, लाई, लागि, ले, वरिपरि, वास्तवमा, वाहेक, विरुद्ध, विशेष, शायद, सँग, सँगै, सक्छ, सट्टा, सधैं, सबै, सबैलाई, समय, सम्भव, सम्म, सही, साँच्चै, सात, साथ, साथै, सायद, सारा, सो, सोध्न, सोही, स्पष्ट, हरे, हरेक, हामी, हामीलाई, हाम्रो, हुँ, हुन, हुने, हुनेछ, हुन्, हुन्छ, हो, होइन, होइनन्, होला, होस् Source “Nepali Stopwords,” n.d.

The output of this step on the aforementioned tweet is as follows:

हेलिकप्टरबाटै, बिउ, दिननै, दर्सैं, मनाउने, व्यवस्थादेखि, पशु, बस्तुलाई, समेत, अस्पतालमा, सुत्केरी, व्यवस्था, मिलाउने, वाचा, डडेल्धुराको, भागेश्वर, गाउँपालिका, वडा, नम्बर, स्वतन्त्रतर्फ, वडाध्यक्षको, उम्मेदवारी, दिएका, खडक, ठकुराठीले, मत, ल्याएका

For this research work, first of all, a count vectorizer i.e. one hot encoding is used however, the number of words in the vocabulary was 114739, and we have a total number of data examples of 88973. So, the shape of our count vectorizer matrix will be 88973×114739, which is problematic in terms of loading data into memory and performing matrix multiplication. That was the main reason to use the TF-IDF feature extraction approach. TF-IDF stands for Term Frequency-Inverse Document Frequency which is a way of converting words into vectors using some numerical statistic. TF-IDF determines the important term in the entire corpus and ignores the less relevant words. Hence, TF-IDF has fewer feature vectors compared to the count vectorizer so, the memory issue can be resolved. For the ease of implementation of TF-IDF, sklearn (Pedregosa et al., 2011) is used.

4.1 Vocabulary

In our research work, we have created a vocabulary as a set of unique words after cleaning the text and tokenizing it. Once the vocabulary is created, we use it to generate the feature matrix using TF-IDF. Some of the words in our vocabulary with corresponding frequencies in a vocabulary are given in the following table.

Table 4.1: Some words in Vocabulary with their frequency.

Nepali Word	**Frequency**
खोयामालेको	1
प्रेरणादायी	1
फाँडेको	1
असहयोगबीच	1
सोंचेझैं	1
थाङ्नोमा	1
लट्पटिए	1

मतलवको	1
केलाउँदै	2
पोल्	2
विज्ञप्तीप्रति	2
बाढियोस्	2
लिटरमा	2
बडदा	2
नचढी	2
वाड्न	2
प्रधानमन्त्री	3007
मतदान	3187
माओवादी	3282
नगरपालिका	3395
भोट	16304

4.2 TF-IDF Features

As explained in the above section, we are using TF-IDF as our main feature extractor. How TF-IDF generates numerical features from text input is described in the section 1.7.3.4. A sample of TF-IDF features for our Upsampled data is in Table 4.2.

Table 4.2: Sample of TF-IDF Feature Matrix.

Feature	जनप्रतिनिधि सर्वोच्च न्यायाधीश महाभियोग हटाउनु	च्याखे थापेर अत्याचार निजामती पास जनप्रतिनिधि झुठो बोलेर करार पास पद
निजामती	0.0	0.345
च्याखे	0.0	0.334
थापेर	0.0	0.336
हटाउनु	0.521	0.0
करार	0.0	0.326
झुठो	0.0	0.318
अत्याचार	0.0	0.314
महाभियोग	0.476	0.0
न्यायाधीश	0.472	0.0
बोलेर	0.0	0.286
सर्वोच्च	0.412	0.0
पास	0.0	0.422
पद	0.0	0.217
जनप्रतिनिधि	0.331	0.207

In the table 4.2, there are two(positive and negative) cleaned sample tweets in the second and third columns. The column feature contains the features and each cell contains the value of TF-IDF. If the particular word not present in sentences, then its corresponding TF-IDF value is 0.

4.3 Analysis and Results

Two separate classes positive and negative were used. First, the system is trained in training datasets in a supervised manner. Then, the system is tested against new samples, that were not used in training, and different performance metrics are measured. The data was partitioned into two parts: training and testing in the 80 to 20 percent ratio. We have a total of 52689 datasets for the positive class and 30229 datasets for the negative class. This is the case of imbalanced class labels. The quantity of the majority class is approximately twice that of the minority class. So major performance matrix in this case is the f1 score (Jeni et al., 2013). In order to generate features, we used TF-IDF. We tried training our model in three different manners. Section 4.4 describes imbalance datasets performance using various ML models similarly section 4.5 describes performance with downsampling, and section 4.6 describes the performance using upsamplling in various ML models.

In order to train logistic regression we used L2 regularization for the penalty, the tolerance for stopping criteria used is 0.0001 with a maximum iteration of 100. In the same way, we implemented the BernoulliNB with α is 1 which is known as the smoothing parameter and the fit-prior value is true which helps to learn class prior probability. Similarly for the SVM kernel used is rbf which is a data matrix with shape (n-sample, n-sample), degree of polynomial kernel function is 3, and gamma value equal to scale is used, and another parameter used was probability estimate which is set false. Furthermore, we train the Decision Tree using the function to measure the quality of a split is gini for gini impurity. The strategy used to choose the split at each node is the best to split. Nodes are expanded until all leaves are pure. Likewise, for Random Forest we used 100 Decision Trees. The function to measure the quality of the split is gini and all nodes are expended until all leaves are pure. In the K Nearest Neighbor Classifier number of neighbors used is 5. All points in each neighborhood are weighted equally and Minkowski distance is used to measure the distance between data points, and in Gradient Boosting log function used to optimize is log loss. The learning rate is 1. The number of boosting stages to perform is 100. The fraction of samples used to fit individual base learners is 1. The function used to measure the quality of split is firedman-mse. The minimum number of samples required to be at a leaf node is 1.

Here, we have taken three articles as baseline models which are mentioned in the table

4.3. However, they had done research on different subject datasets like aspect-related datasets (which are focused on particular products, and things) and COVID-19-related tweets with low amounts. They implemented TF-IDF as feature extraction. Furthermore, authors have used the value of parameters and hyperparameters according to best for their model. The remaining working principles like tokenization, data cleaning, stop words removal, and feature extraction process were the same as ours.

The summary performance of sentiment classification of some previous studies on tweets is displayed in the following table 4.3

Table 4.3: Previous studies model evaluation

Research	**Classifiers**	**F1-Score**	**Accuracy**
Tamrakar et al., 2020	SVM BNB	76.9% 77.7%	76.8% 77.5%
Sitaula et al., 2021	SVM RF LR XG Boost K-NN SVM RBF	62.2% 67.5% 67.4% 69.5% 65.2% 51%	63.9% 63.5% 64.7% 66.7% 60.3% 40.2%
Shrestha and Bal, 2020	SVM RF Decision Tree	80.2% 77% 68.2%	- - -

4.4 Model Performance on Imbalance Datasets

In this experiment, we trained the model without performing anything to overcome the imbalance. The following table explains the results we observed for the testing datasets. The overall performance of all the models is presented in Table 4.4.

Table 4.4: Testing Performance with Imbalanced Data.

Classifier	**Class**	**Precision**	**Recall**	**F1-Score**	**Accuracy**
	Negative	0.77	0.87	0.82	
Logistic Regression					0.7533

	Positive	0.71	0.55	0.62	
	All	0.7106	0.5533	0.6222	
Naive Bayes	Negative	0.77	0.81	0.79	0.7281
	Positive	0.64	0.60	0.62	
	All	0.63923	0.5951	0.6164	
SVM	Negative	0.78	0.85	0.81	0.7547
	Positive	0.70	0.59	0.64	
	All	0.6969	0.5869	0.6372	
Decision Tree	Negative	0.74	0.78	0.76	0.6890
	Positive	0.58	0.54	0.56	
	All	0.5826	0.5387	0.5598	
Random Forest	Negative	0.74	0.89	0.81	0.7381
	Positive	0.70	0.47	0.56	
	All	0.7038	0.4660	0.5608	
SVM RBF	Negative	0.78	0.84	0.81	0.7501
	Positive	0.68	0.60	0.64	
	All	0.6809	0.6005	**0.6382**	
K-NN	Negative	0.72	0.86	0.79	0.6746
	Positive	0.64	0.42	0.50	
	All	0.6412	0.4162	0.5048	
Gradient Boosting	Negative	0.66	0.99	0.79	0.6746
	Positive	0.87	0.13	0.23	

	All	0.8713	0.1331	0.2310	
Ada Boost	Negative	0.64	1	0.78	0.6444
	Positive	0.90	0.40	0.7	
	All	0.8957	0.40	0.06	
XG Boost	Negative	0.72	0.91	0.80	0.7137
	Positive	0.71	0.38	0.49	
	All	0.7118	0.3786	0.4943	

On the basis of the performance table, SVM with the RBF kernel gives the best result on the imbalance dataset with an overall performance of F1-Score of 63.82%. Our model performed 63.82% well in terms of minimizing false positives and false negatives. Similarly, in class levels, its performance is 81% for negative class and 56% for positive class. The reason behind this performance is sentiment classification tasks involve non-linear relationships in the feature space, and the RBF kernel has the capacity to capture complex, non-linear decision boundaries. This kind of flexibility allows the RBF kernel to model patterns in sentiment-related features. Similarly, the RBF kernel implicitly maps the input features into a higher-dimensional space. This behavior can be beneficial when the sentiment-related features are not easily separable in the original feature space. Furthermore, secondly, SVM has better performance of having 63.72% F1-score in overall.

4.5 Model Performance with Downsample

One of the major problems associated with classification problems is data imbalance. Downsampling (Lee and Seo, 2022) is one of the approaches to handling the problem of data imbalance. With an imbalanced dataset performance of the classification algorithm decreases as most of the dataset is prone toward the majority class. In this method we decrease the amount of the majority class; i.e. positive class equal to the minority class. We have 30229 datasets of each of the classes. (García et al., 2009)

Once we have a balanced dataset, we use accuracy as the major performance matrix. Accuracy measures the correct prediction made by the model. The overall performance of all models is demonstrated in the Table 4.5.

Table 4.5: Testing Performance with Down-sampled Data.

Classifier	Class	Precision	Recall	F1-Score	Accuracy
Logistic Regression	Negative	0.75	0.71	0.73	**0.7402**
	Positive	0.73	0.77	0.75	
	All	0.7286	0.774	0.7509	
Naive Bayes	Negative	0.73	0.73	0.73	0.7301
	Positive	0.73	0.73	0.73	
	All	0.7325	0.7341	0.7333	
SVM	Negative	0.74	0.71	0.73	0.7363
	Positive	0.73	0.76	0.74	
	All	0.7363	0.7609	0.7447	
Decision Tree	Negative	0.64	0.66	0.65	0.6561
	Positive	0.66	0.63	0.65	
	All	0.6590	0.6341	0.6463	
Random Forest	Negative	0.72	0.69	0.71	0.7181
	Positive	0.71	0.74	0.72	
	All	0.7034	0.7392	0.7244	
SVM RBF	Negative	0.73	0.71	0.72	0.7363
	Positive	0.73	0.75	0.74	

Table 4.5: Testing Performance with Down-sampled Data.

Classifier	Class	Precision	Recall	F1-Score	Accuracy
	All	0.7265	0.7452	0.7447	
K-NN	Negative	0.63	0.77	0.69	0.6651
	Positive	0.71	0.57	0.63	
	All	0.7124	0.5660	0.6308	
Gradient Boosting	Negative	0.71	0.50	0.59	0.6543
	Positive	0.62	0.80	0.70	
	All	0.6211	0.8014	0.6998	
Ada Boost	Negative	0.72	0.35	0.47	0.6122
	Positive	0.58	0.87	0.69	
	All	0.5771	0.870	0.6942	
XG Boost	Negative	0.71	0.62	0.66	0.6869
	Positive	0.67	0.76	0.71	
	All	0.6679	0.7571	0.7097	

While observing the performance of all implemented algorithms Logistic Regression has the highest accuracy with an overall value of 75.09%. Our logistic Regression performs 75.09% correctly. The reason is Logistic Regression considers the linear relationship between the input features and the log-odds of the output. In most sentiment classification tasks, especially when using bag-of-words and TF-IDF the relationship between features and sentiment can be modeled as linear. LR provides a clear interpretation of model coefficients. Each of the coefficients represents the impact of corresponding features on the log odds of the predicted outcome. This type of interpretation is valuable in understanding features that are effective in classifying sentiment.

4.6 Model Performance with Upsample

Another approach we implemented in this research work is upsampling. (Bae et al., 2021) Upsampling is the process of increasing the amount of minority class i.e. in our case negative class equal with the majority class. We have a total of 52689 datasets for positive class and we increase the quantity of negative class in equal amount. However, being an equal amount of dataset in two labels we have chosen f1-score as the major performance matrix because there is the chance of being overfitting of the dataset in the minority class. Among all trained models performance of Logistic Regression is best in this case. We have a large amount of data a total of 105,378 datasets. As we know performance of ML models increases as datasets increase. The overall performance of all models is displayed in Table 4.6.

Table 4.6: Testing Performance with Up-sampled Data.

Classifier	Class	Precision	Recall	F1-Score	Accuracy
Logistic Regression	Negative	0.87	0.82	0.84	0.8476
	Positive	0.83	0.88	0.85	
	All	0.8290	0.8760	**0.8518**	
Naive Bayes	Negative	0.77	0.72	0.74	0.7505
	Positive	0.74	0.78	0.76	
	All	0.7369	0.7774	0.7566	
SVM	Negative	0.80	0.74	0.77	0.7788
	Positive	0.76	0.81	0.78	
	All	0.7589	0.8073	0.7848	
Decision Tree	Negative	0.83	0.72	0.77	0.7869
	Positive	0.75	0.85	0.80	

	All	0.7542	0.8496	0.799	
Random Forest	Negative	0.86	0.75	0.80	0.8170
	Positive	0.78	0.88	0.83	
	All	0.7799	0.8819	0.8278	
SVM RBF	Negative	0.80	0.75	0.77	0.7788
	Positive	0.76	0.81	0.78	
	All	0.7624	0.8085	0.7848	
K-NN	Negative	0.81	0.77	0.79	0.7195
	Positive	0.78	0.82	0.80	
	All	0.7780	0.8210	0.7989	
Gradient Boosting	Negative	0.71	0.49	0.58	0.6451
	Positive	0.61	0.80	0.69	
	All	0.6105	0.7970	0.6914	
Ada Boost	Negative	0.75	0.31	0.44	0.5932
	Positive	0.56	0.90	0.69	
	All	0.5647	0.8956	0.6926	
XG Boost	Negative	0.73	0.62	0.67	0.6976
	Positive	0.67	0.77	0.72	
	All	0.6713	0.7714	0.7179	

From the performance table above Logistic Regression performs better than other classification algorithms with 85.18% f1-score, 84.76% f1-Score, 87.60% recall, and 82.90% precision. Secondly, Random Forest has the second-best performance of 82.78%

f1-score, 81.70% accuracy, 88.19% recall, and 77.99% precision. Our best-performing model outperforms the baseline results of prior models on related studies.

From the results obtained from three different variations of the dataset i.e. up-sampling, down-sampling, and no-sampling we found the best result using upsampling. One of the possible reasons is that machine learning models are data-hungry models and by doing upsampling, we provided enough data for both classes. Here, we found Logistic Regression and Random Forest performing better than baseline models as well.

Some of the reasons why Logistic Regression performs well in our study are demonstrated in the following points,

- LR takes the log odds of the probability of a positive class as a linear combination of input features. This boundary can be beneficial in capturing the relation between features and sentiment in up-sampled data.
- LR assigns less weight to unwanted or less important features. This can be effective when working with upsampled data, as it helps the classifier focus on the most important features related to the sentiment.
- LR implements the concept of regularization. Which helps to penalize the large coefficients and prevent the model from overfitting. As in upsampling, we increase the instances of the minority class so prevention from overfitting is crucial.
- LR is computationally effective and can handle large datasets.

4.7 Parameters to classify tweets sentiment

We found some parameters that help us to classify the positive and negative sentiments are described in the following subsection.

4.7.1 Sentiment bearing words

Sentiment-bearing words are words in pieces of text that express sentiment and emotion. For sentiment analysis identifying sentiment-bearing words is important to determine the overall sentiment expressed in a piece of text. Below are some examples of sentiment-bearing words we found in our dataset for positive and negative sentiments,

- **Positive:** जनचेतना, सहकारीता, सशक्तीकरण, विकास, स्वतन्त्रता, समृद्धि, न्याय, सम्मान, प्रगति, सम्झौता, सम्बन्ध
- **Negative:** भ्रष्टाचार, बेइमानी, बहिष्कार, अस्वीकृति, अन्याय, विघात, दुःख, दबाब, विद्रोह

4.7.2 Negation words/Double Negation Words

Negation words are words that point to the negation or denial of something in a sentence. They are commonly used to express the absence or opposite of something. We found negation words are another crucial parameter in determining the sentiment of pieces of text. In the same way, double negation is the words that contain double negative words. Some of the examples of negation-bearing words we found in our dataset are given below.

- **Negation Words:** होइन, नभए, न, नगरे, नहुँदा, भएन, नहुँदै, नरहेको, नपुगेको, नपाइएको
- **Double Negation Words:** कहिले पनि होइन, कुनै छैन, कसैले पनि गरेको छैन, कसैले पनि नदेखेको छैन, कहिले पनि देखेको छैन

4.7.3 TF-IDF

Term frequency-inverse document frequency is all about finding the occurrence of words in a particular document either a positive document or a negative document. The frequency of particular terms(words) in specific classes (positive, negative) is another parameter to classify the sentiment of text.

4.7.4 n-grams

n-grams are the sequences of n-words or tokens that we implement to find out the emotional tone of a piece of text. n-grams is used to describe the sequence of adjacent words. In the context of the n-gram n variable, that may take positive integer 1,2,3,...n. n-grams capture the contextual information and relationships between words in the text, which is crucial for sentiment analysis. Unigrams are individual words or tokens taken from the text. Each word is treated as a separate feature. Unigrams are the simple form of n-grams used in sentiment analysis. Furthermore, in bigrams, pairs of adjacent or consecutive words or tokens were taken from the text. They give the

relationships between nearby words. Bigrams can provide more context than unigrams and help to correctly identify the sentiment of the text. In the same way, trigrams are sequences of three adjacent words taken from the text. They can capture more complex relations between words and can deliver deeper meaning than unigrams and bigrams.

Some examples of unigrams, bigrams, and trigrams from our dataset are explained in the following points.

- **Unigrams:** नेपाल, राजनीति, पार्टी, नेता, सरकार, विपक्ष, निर्वाचन, संघर्ष, आन्दोलन, जनमत, प्रधानमन्त्री
- **Bigrams:** नेपाली राजनीति, राजनीतिक पार्टी , नेता संगठन, सरकारी निर्णय, निर्वाचन आयोग, राजनीतिक सम्मेलन
- **Trigrams:** नेपालको राजनीतिक स्थिति, नेपाली राजनीतिक दल, सरकारी निर्णय नेपाल, नेपाली राजनीतिक प्रवृत्ति, नेपालको राजनीतिक समस्या, नेपाली राजनीतिक जीवन, नेपाली राजनीतिक संगठन

5 CONCLUSION

5.1 Conclusion

In this research work, we prepared a dataset for sentiment classification. Sentiment classification with different supervised machine learning algorithms is presented and evaluated on Nepali political tweets. The important preprocessing steps and feature extraction techniques i.e. TF-IDF for sentiment classification are also described in full detail. Furthermore, experiments were done by handling imbalanced data by different approaches.

Sentiment classification using Logistic Regression with upsampling methods gives the best performance of **85.18%** f1-score with accuracy **84.76%**, precision **82.90%**, and recall **87.60%** respectively. Furthermore, looking at the downsampling approach Logistic Regression again performs well with **74.02%** accuracy, **75.09%** f1-score, **77.4%** recall, **72.86%** precision. This numerical figure shows that our best-performing model outperforms the baseline results of prior models on related studies. Similarly, Random Forests and Decision Trees also perform well. While using the downsampling Logistic regression also has the best performance with **74.02%** Accuracy. However, we found weak performance in the imbalanced dataset. Furthermore, parameters that help to classify the sentiment of tweets are sentiment-bearing words, negation words, double negation words, TF-IDF, and n-gram.

Sentiment classification is a difficult problem, not only because of the great number of variations in human language but also because of the words used to write text with a variety of communities and social media platforms. The difficulty also adds up while collecting data and labeling it manually. As labeling was done by a single person, the labels might be biased as well.

5.2 Recommendation for Future

Considering the difficulties in handling the imbalanced data and the risk of biases in labeling, this research work will get better results if multiple persons label the datasets. Furthermore, using different sources of data(i.e. only tweets) might also increase the performance. As this research has been done based on the TF-IDF, and traditional machine learning algorithms, there are various topics to research on. A large training corpus with label class trained with a deep learning model with an advanced feature extraction method can help in good generalization of the system. The future scope of this research can be as follows,

- The proposed system can be trained further by collecting more data and making balanced datasets.
- The proposed system can be extended for the recognition of the sentiment of words, sentences, and documents.
- The proposed system can be extended for multi-class classification systems by adding new labels like neutral.
- The proposed system can be trained with feature extraction methods like word2vec and Countvectorizer.
- Instead of traditional machine learning algorithms, we can also train deep learning algorithms like LSTM, and GRU.
- There has been various research in NLP using state-of-the-art algorithms like Transformers and its variants. Sentiment classification can get better results using these algorithms.

REFERENCES

Adhikari, R., Thapaliya, S., Basnet, N., Poudel, S., Shakya, A., & Khanal, B. (2022). COVID-19-related Nepali tweets classification in a low resource setting. *Proceedings of The Seventh Workshop on Social Media Mining for Health Applications, Workshop & Shared Task*, 209–215. https://aclanthology.org/2022.smm4h-1.52

Akhmetov, I., Gelbukh, A., & Mussabayev, R. (2022). Topic-aware sentiment analysis of news articles. *Computacion y Sistemas*, *26*, 423–439. https://doi.org/10.13053/CyS-26-1-4179

Bae, S.-Y., Lee, J., Jeong, J., Lim, C., & Choi, J. (2021). Effective data-balancing methods for class-imbalanced genotoxicity datasets using machine learning algorithms and molecular fingerprints. *Computational Toxicology*, *20*, 100178. https://doi.org/https://doi.org/10.1016/j.comtox.2021.100178

Confusion matrix [2024]. (n.d.).

DataReportal. (2023). *Digital 2023: Nepal*. https://datareportal.com/reports/digital-2023-nepal (accessed: 01.05.2023).

de Souza, G. F. M., Caminada Netto, A., de Andrade Melani, A. H., de Carvalho Michalski, M. A., & da Silva, R. F. (2022). Chapter 6 - engineering systems' fault diagnosis methods. In G. F. M. de Souza, A. Caminada Netto, A. H. de Andrade Melani, M. A. de Carvalho Michalski, & R. F. da Silva (Eds.), *Reliability analysis and asset management of engineering systems* (pp. 165–187). Elsevier. https://doi.org/https://doi.org/10.1016/B978-0-12-823521-8.00006-2

Evgeniou, T., & Pontil, M. (2001). Support vector machines: Theory and applications. *2049*, 249–257. https://doi.org/10.1007/3-540-44673-7_12

G, T. R., Sitaula, C., Basnet, A., Mainali, A., & Shahi, T. B. (2021). Deep learning-based methods for sentiment analysis on nepali covid-19-related tweets. *Computational Intelligence and Neuroscience*, *2021*, 2158184. https://doi.org/10.1155/2021/2158184

García, V., Mollineda, R., & Sánchez, J. (2009). Index of balanced accuracy: A performance measure for skewed class distributions. *5524*, 441–448. https://doi.org/10.1007/978-3-642-02172-5_57

Jeni, L., Cohn, J., & De la Torre, F. (2013). Facing imbalanced data - recommendations for the use of performance metrics. *Proceedings - 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction, ACII 2013*, *2013*. https://doi.org/10.1109/ACII.2013.47

Kadhim, A. (2018). An evaluation of preprocessing techniques for text classification. *International Journal of Computer Science and Information Security, 16*, 22–32.

Lee, W., & Seo, K. (2022). Downsampling for binary classification with a highly imbalanced dataset using active learning. *Big Data Research*, *28*, 100314. https://doi.org/https://doi.org/10.1016/j.bdr.2022.100314

Nepali stopwords [Accessed: [Date]]. (n.d.).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

ScienceDirect. (2024). Logistic regression analysis.

Shahi, T., & Pant, A. (2018). Nepali news classification using naïve bayes, support vector machines and neural networks, 1–5. https://doi.org/10.1109/ICCICT.2018.8325883

Shrestha, B. B., & Bal, B. K. (2020). Named-entity based sentiment analysis of nepali news media texts. *NLPTEA*. https://api.semanticscholar.org/CorpusID:227905372

Singh, O. M. (2019). Nepali multi-class text classification.

Singh, O. M., Timilsina, S., Bal, B. K., & Joshi, A. (2020). Aspect based abusive sentiment detection in nepali social media texts. *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 301–308. https://doi.org/10.1109/ASONAM49781.2020.9381292

Sitaula, C., Basnet, A., Mainali, A., & Shahi, T. B. (2021). Deep learning-based methods for sentiment analysis on nepali covid-19-related tweets. *Computational In-*

telligence and Neuroscience, *2021*, 2158184. https://doi.org/10.1155/2021/2158184

Tamrakar, S., Bal, B. K., & Thapa, R. B. (2020). Aspect based sentiment analysis of nepali text using support vector machine and naive bayes. *Technical Journal*, *2*(1), 22–29. https://doi.org/10.3126/tj.v2i1.32824

Wikipedia contributors. (2024). Decision tree learning — Wikipedia, the free encyclopedia [[Online; accessed 11-February-2024]]. https://en.wikipedia.org/w/index.php?title=Decision_tree_learning&oldid=1203415793

A Dataset and Sample Source Codes

All source codes and datasets used in this research are available in the respective links.

- Source Code
- Datasets