Sentiment Analysis for Assassination of Imran Khan in Twitter

Ajay Kumar T and Vasanth Ramm P K
Department of Information Technology
Mepco Schlenk Engineering College
Sivakasi,India
Akvnr2152002@mepcoeng.ac.in

Raja Athiban P
Department of Information Technology
Mepco Schlenk Engineering College
Sivakasi,India
athiban.p2015_it@mepcoeng.ac.in

Mrs. M.Blessa Binolin Pepsi
Assistant Professor
Department of Information Technology
Mepco Schlenk Engineering College
Sivakasi,India

Dr J.Maruthu Pandi
Assistant Professor
Department of Information Technology
Mepco Schlenk Engineering College
Sivakasi,India

***Abstract* - With the rapid increase in internet usage, sentiment analysis has become one of the most popular areas of natural language processing (NLP). Using sentiment analysis, the implied emotions in the text can be efficiently mined for different occasions. Social media is being widely used by people to receive and transmit various kinds of information. Mining such content to gauge people's feelings can play a vital role in the decision to keep the situation under control. The aim of this study is to elicit the sentiments of Assassination of Imran khan tweets in the twitter posted by different peoples. In this work, sentiment analysis of tweets sent by every citizens was done using NLP and machine learning classifiers. A total of 24,011 tweets containing the keywords "Firing" were extracted. Data was extracted from Twitter using the Twint, annotated using the TextBlob and VADER lexicons, and preprocessed using the natural language toolkit provided by Python. Eight different classifiers were used to classify the data. The experiment achieved the highest accuracy of 98.4% with the LinearSVC classifier and unigrams. This study concludes that the majority of people have posted netural tweets, some of them have positive tweets and only few percentange of people has posted negative tweets.**

***Index Terms* - Machine learning, Imran Khan, NLP, and firing.**

I. INTRODUCTION

On 3 November 2022, Imran Khan, former prime-minister of Pakistan and president of the Pakistan Tehreek-e-Insaf political party, was shot in an assassination attempt in Wazirabad, Punjab, during a long kick march against the government. The marksman also injured a number of other PTI leaders and killed a supporter.

Some of the domestic and international reaction for this assassination attempt as follows

Prime Minister Shehbaz Sharif condemned the attack on Khan" in the strongest words", soliciting for the recovery of Khan and the other people who were injured, and saying," I've directed Interior Minister for an immediate report on the incident."He also said that the civil government would give all necessary support to the parochial government in Punjab in security and disquisition.

The Pakistani service condemned the attack and offered" sincere prayers for precious life lost and speedy recovery of Khan".

Egypt- The Ministry of Foreign Affairs condemned the tried assassination and wished Khan a speedy recovery from his injuries in a statement.

India- A prophet for the Ministry of External Affairs said that his country is" nearly keeping an eye" on developments in Pakistan.

Iran- Ministry of Foreign Affairs prophet Nasser Kanaani condemned the assassination attempt against Khan and wished a speedy recovery.

United Arab Emirates- The Ministry of Foreign Affairs condemned the assassination attempt against Khan. In a statement, it said," UAE expresses its strong combination of these lawless acts that aim to destabilize security and stability and are inconsistent with philanthropic values and principles."

After this assassination attempt, people use Twitter to partake their opinions. Also, there's limited knowledge of the general public's sentiment about the main motifs which are in discussion over time. The sentiment categorization can be enforced by using various approaches. We can majorly classify these approaches in the following three types 1) lexicon - based approach, 2) machine learning / deep learning approach, and 3) hybrid approach. We've used a machine literacy approach to classify the opinion of Pakistan's common public in this work. To use machine literacy for sentiment analysis, data pre-processing on raw data is a prerequisite since the effectiveness of the algorithm used is directly commensurable to the quality of training and testing dataset. In sentiment analysis, the pre-processing of textbook is known as natural language processing (NLP). It includes the following six way depicted in Fig. 1.

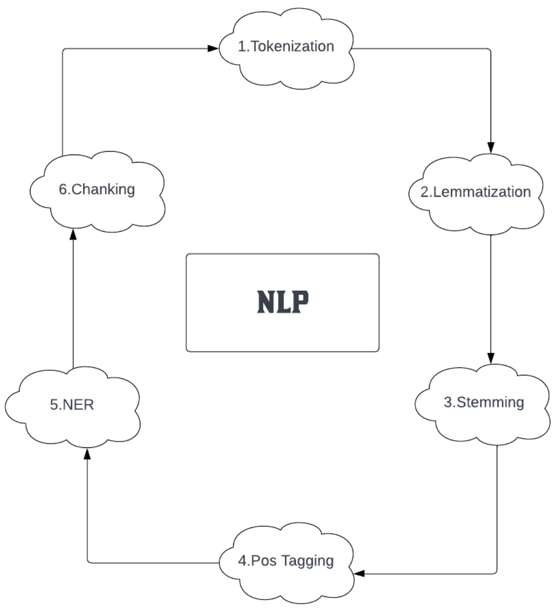

Fig. 1. Steps of NLP.

Sentiment analysis, is also called as sentiment mining, is a natural language processing (NLP) approach that identifies the emotional tone behind the body of text. NLP determine and categorize opinions about a service, product or idea. Sentiment analysis focuses on text polarity (positive, negative, neutral), but also goes beyond polarity to reveal specific feelings and emotions (angry, happy, sad, etc.) and even intent. Natural Language Processing has emotion detection systems which uses lexicons or complex machine learning algorithms.

Natural language processing (NLP) defines the branch of computer technology—and greater, the department of artificial intelligence or AI—involved with giving computer systems the potential to understand textual content and spoken phrases in an awful lot the same way humans can.

The library used in this project are pandas, text blob, nltk, count vectorizer,matplotlib.

II. PROPOSED APPROACH

In this section, proposed a framework for sentiment analysis of assassination of Imran khan. The framework is shown in Figure 2. The various phases of the framework to perform the sentiment analysis of assassination of Imran khan are as follows

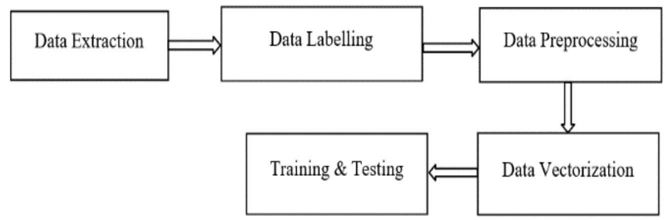

Fig. 2. Methodology used

A. DATA EXTRACTION:

Twint is an open-source python library for accessing the Twitter API. It gives you an interface to access the API from your Python application. In this project, we used the tweepy to extract the tweets from twitter for the keyword "Firing".

B. DATA LABELING:

After tweets assortment, we've used the subsequent approach shown in below to label the tweets as positive, neutral, and negative. we've generated every tweet's polarity victimisation the TextBlob library and VADER (Valence Aware wordbook for sEntiment Reasoning) tool of the Python. Next, we've taken the intersection of Text Blob and VADER results to consolidate the polarities.

A Python (2 and 3) package called Text Blob is used to process textual data. It offers a straightforward API for getting started with typical natural language processing (NLP) activities like part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and others.

The tweets are categorised using this method by textblob, which employs a polarity with a range of -1 to +1. The tweets are divided into positive, negative, and neutral categories by the polarity.

VADER (Valence Aware wordbook and sentiment Reasoner) could be a lexicon and rule-based sentiment analysis tool specifically designed for social media sentiments. Vader is optimized for social media information and may offer smart results once used with information from Twitter, Facebook, etc. VADER uses a mixture of a sentiment lexicon, a listing of lexical options (e.g., words) that area unit typically labeled as positive or negative in keeping with their linguistics orientation. VADER provides data not solely regarding the positivism and negativity worth, however additionally regarding however positive or negative a sentiment is.

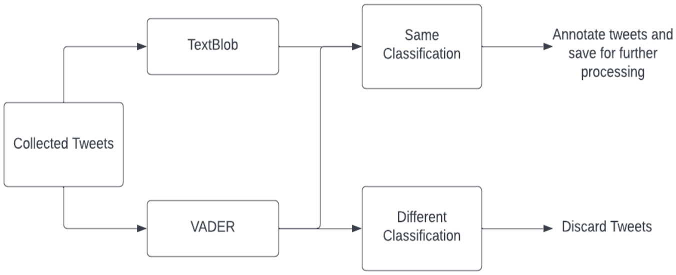

Fig. 3. Data Labelling

D. DATA PREPROCESSING:

Natural Language ToolKit, also known as NLTK, is a Python toolkit designed for use with natural language processing. It offers us a variety of text processing libraries and a large number of test datasets. Using NLTK, a range of operations, including tokenizing and visualising parse trees, may be completed.

Data collected may hold some unsought and sentiment fewer words like links, Twitter-specific words like hashtags (starts with #) and tags (starts with @), single letter words, numbers, etc. These types of words can play the role of noise in our classifier work and testing. To amend classifier efficiency, it is necessary to induce obviate noise from the labelled info set before feeding the classifier. Our pre-processing module separates noise from the labelled info set. The steps of pre-processing area unit shown below. throughout this step, we have a tendency to tend to implement a module to induce obviate the above-specified impurities, born-again set into an information frame, then dead the removal of English stop words, string punctuation, tokenization, stemming, and lemmatization.

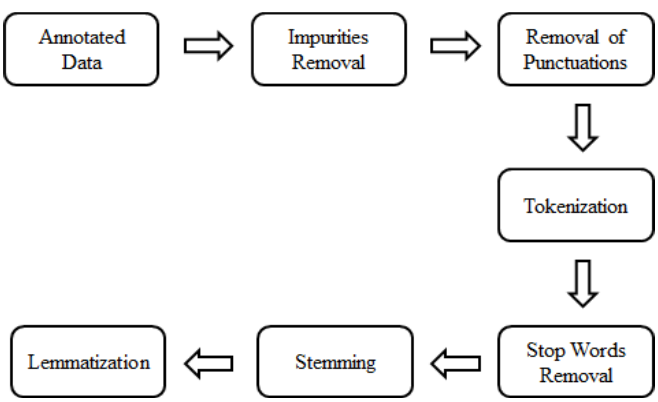

Fig. 4. Data pre-processing.

Tokenization - The process of breaking up a big sample of text into words is called word tokenization. This is necessary for jobs involving natural language processing, where each word must be recorded and submitted to additional analysis, such as classification and counting for a specific sentiment, etc.

Stemming - By reducing word inflection to its root forms, a process known as stemming, it is possible to map a group of words to a single stem even when the stem is not a legitimate word in the language.

Lemmatization is the process of combining a word's several inflected forms into a single unit for analysis. Similar to stemming, lemmatization adds context to the words. As a result, it ties words with related meanings together.

E. VECTORIZATION:

The scikit-learn Python library offers a fantastic feature called Count Vectorizer. It is employed to convert a given text into a vector based on the frequency (count) of each word that appears throughout the text.

The machine learning classifiers cannot take the input written in any language except numbers. Thus, before mistreatment the text knowledge for prognostic modeling, it's needed to convert it into options. we've used the Count Vectorizer feature extractor to calculate word frequencies. Count Vectorizer counts the frequency of every word gift within the document and creates a thin matrix, as shown below. as an example, Doc1: “She was young the way an actual young person is young .” Count Vectorizer can convert this text into the subsequent thin matrix with associate index of the words in alphabetical order as follows: four, “was”: 6, “young”: 8, “the”: 5, “way”: 7, “an”: 1, “actual”: zero, “person”: three, “is”: 2}.

This matrix isn't thin as a result of we tend to square measure changing the sole single document. within the case of multiple documents, it's frequent that a word gift in one document will be missing from another documents, and therefore the corresponding cells square measure crammed up with zero, and therefore the resultant matrix can become thin.

SAMPLE MATRIX BY COUNT VECTORIZER

Index	0	1	2	3	4	5	6	7	8
Doc1	1	1	1	1	1	1	1	1	3

Fig. 5. Count Vectorization

F. TRAINING AND TESTING:

After feature extraction of the preprocessed knowledge set, we've got passed the information to machine learning classifiers. we've got used eight classifiers (Multinomial Naive Bayes, Bernoulli Naïve Bayes, Logistic Regression, Linear SVC, AdaBoostClassifier, Ridge Classifier, Passive Aggressive Classifier and Perceptron) for this purpose. we've got used eightieth knowledge for coaching and 2 hundredth knowledge for testing the classifiers. we've got extracted the performance of the classifiers mentioned on top of mistreatment 1-g, 2-g, and

3-g. The detailed description about classifiers used for training and testing the model as follows.

The Multinomial Naive Bayes algorithm is a popular Bayesian learning algorithm. Based on the Bayes theorem, the program guesses the tag of a text, such as an email or a newspaper article. For a given sample, it calculates the likelihood of each tag and outputs the tag with the highest probability. Multinomial Naive Bayes classifiers are suitable for classification using discrete features (e.g., word counts for text classification). Numbers of features are normally integers in a multinomial distribution.

$$P(A|B) = P(A) * P(B|A)/P(B)$$

When predictor B is already known, we are computing the probability of class A.
P(B) = prior probability of B
P(A) = prior probability of class A
P(B|A) = occurrence of predictor B given class A probability

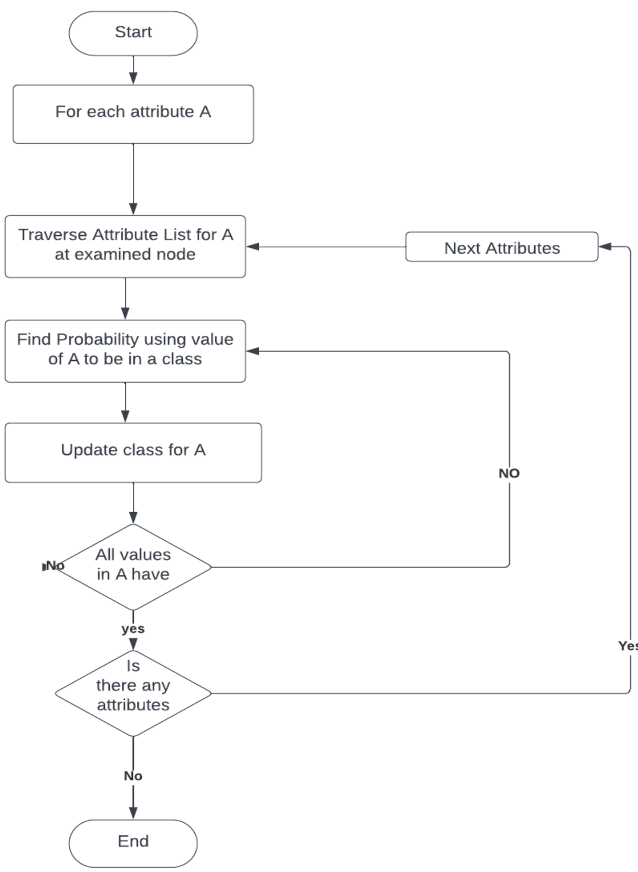

Fig. 6. Workflow for Multinomial Naïve Bayes

For data that is distributed according to multivariate Bernoulli distributions, Bernoulli NB implements the naive Bayes training and classification algorithms. There may be numerous features, but each one is assumed to be a binary-valued (Bernoulli, Boolean) variable. Bernoulli The Naive Bayes family includes Naive Bayes. It accepts just binary values. The most basic example is when we determine whether or not a word will appear in a document for each value. That model is quite condensed. When counting word frequencies is less crucial, Bernoulli might get more accurate findings. Simply put, we must count each value for the binary term occurrence features, which determine if a word appears in a document or not. Instead of determining a word's frequency within the document, these features are utilized.

Let there be a variant 'X' and let the likelihood of success be denoted by 'p' and also the probability of failure be delineate by 'q.'

q = 1 - (Success probability)
q = 1 - p

Logistic regression is one amongst the foremost well-liked Machine Learning algorithms, that comes underneath the supervised Learning technique. it's used for predicting the explicit variable quantity employing a given set of freelance variables. supply regression predicts the output of a categorical variable quantity. supported the number of classes, supply regression will be classified as:

Binomial: target variable will have solely two potential types: “0” or “1” which can represent “employee” vs “unemployee”, “yes” vs “no”, etc.

Multinomial: target variable will have three or a lot of potential sorts that don't seem to be ordered (i.e., sorts haven't any quantitative significance) like “disease A” vs “disease B” vs “disease C”.

Ordinal: it deals with target variables with ordered classes. for instance, a take a look at score will be categorised as: “very poor”, “poor”, “good”, “very good”. Here, every class will be given a score like zero, 1, 2, 3.

The supply regression of y on x will be obtained from the simple regression equation. The mathematical steps to induce supply Regression equations area unit given below:

we all know the equation of the line will be written as:

$$y = b\_0 + b\_1\, x\_1 + b\_2\, x\_2 + \cdots + b\_n\, x\_n$$

Let's divide the preceding equation by (1-y) because y in Logistic Regression can only be between 0 and 1 in order to account for this:

$y/(1-y)$; 0 for y=0 and infinity for y=1

However, we require a range between -[infinity] and+[infinity]. If we take the equation's logarithm, it becomes:

$$log\ y/(1-y) = b\_0 + b\_1\ x\_1 + b\_2\ x\_2 + \cdots + b\_n\ x\_n$$

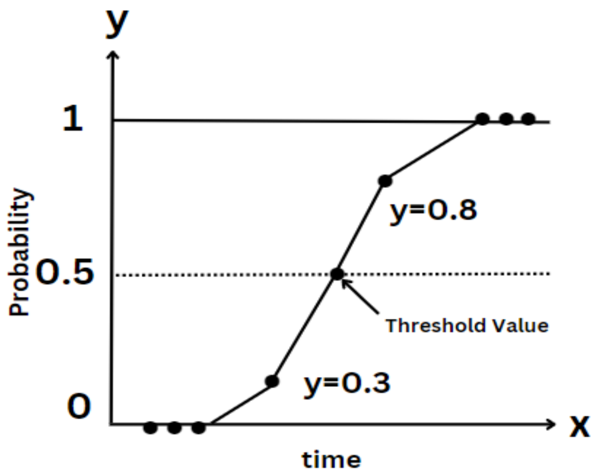

Fig. 7. Sigmoid Function in Logistic Regression

Support vector classifiers, or linear SVCs, split or categorise your data by returning a "best fit" hyperplane that fits to the data you supply. You may then feed some features to your classifier to get the "predicted" class after acquiring the hyperplane. A technique called the Linear Support Vector Machine (Linear SVC) looks for a hyperplane to maximise the distance between samples that are classified. With a high number of data, the Linear Support Vector Classifier (SVC) approach performs well. It uses a linear kernel function to perform classification. When compared to the SVC model, the Linear SVC adds more parameters including the loss function and penalty normalisation, which applies "L1" or "L2." Linear SVC is based on the kernel linear technique, the kernel method cannot be modified.

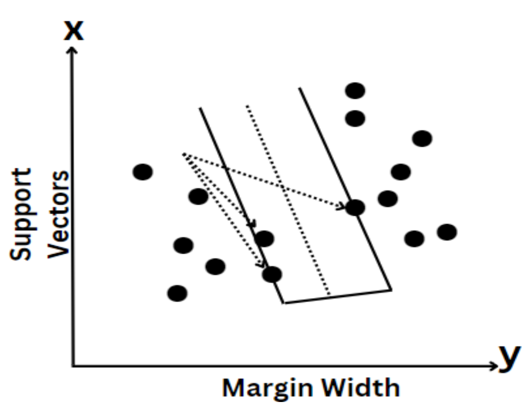

Fig. 8. Support Vectors & Margin Width

To create a strong classifier, the Ada-boost classifier combines weak classifier algorithms. One algorithm might not classify the objects well enough. However, we can achieve a reasonable accuracy score for the overall classifier if we combine many classifiers with the choice of the training set at each iteration and the proper amount of weighting in the final voting.

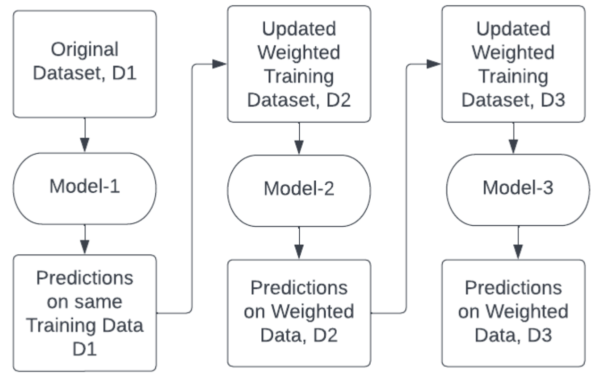

Fig. 9. Workflow of AdaBoost Classifier

$$\alpha = 1/2 * \ln ((1 - E)/E)$$

The classifier's weight is straightforward; it is determined by the error rate E. Initial weighting for each input training example is equal.

A Ridge regressor is essentially a Linear Regressor that has been regularized. In other words, we add a regularized term to the initial cost function of the linear regressor in order to drive the learning algorithm to suit the data and help maintain the weights as low as feasible. The regularized term's 'alpha' parameter regulates the model's regularization, hence lowering the variance of the estimations. the target variable is suitably transformed into +1 and -1. Create a Ridge model using a mean square loss function and L2 regularization (ridge) as the penalty term.

If the anticipated value is less than 0, the class label is predicted to be -1; otherwise, the class label is predicted to be +1. Ridge classification is a method used in machine learning to examine linear discriminant models.

The passive aggressive classifier algorithm, which belongs to the class of online learning algorithms, is capable of handling enormous datasets and alters its model in response to each new instance it meets. A family of machine learning algorithms known as passive-aggressive algorithms is frequently utilised in large data applications. Large-scale learning typically uses passive-aggressive algorithms. It is one of the algorithms used in online learning.

Passive: Maintain the model and make no changes if the prediction is accurate. In other words, the example's data are insufficient to alter the model in any way.

Aggressive: Modify the model if the prediction turns out to be inaccurate. In other words, a model modification could make it right.

C: The model's penalization for making erroneous predictions is indicated by this regularization parameter.

max iter: The most iterations the model does on the training set of data.

A machine learning technique called a perceptron imitates the functioning of a neuron in the brain. It is also known as an one neuron, one layer neural network. The result of just one activation function connected to a single neuron determines the output of this neural network.

$$\sum w\_i\, x\_i = w\_1\, x\_1 + w\_2\, x\_2 \;+\ldots+ w\_n\, x\_n$$

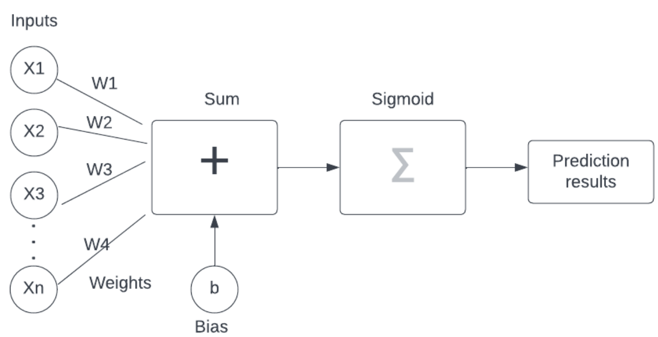

Fig. 10. Workflow for Perceptron.

The weight coefficient is automatically learned in a perceptron. Weights are first multiplied by input features to determine whether to fire the neuron or not. The hard limit transfer function of a perceptron limits the output to a binary number (0 or 1) alone. Only sets of input vectors that can be linearly separated can be classified using perceptron's. Non-linear input vectors are difficult to properly categorize.

Ensemble learning is a method for solving specific computational intelligence problems by carefully generating and combining a number of models, such as classifiers or experts. The main purpose of ensemble learning is to raise (classification, prediction, function approximation, etc.).

A random forest is a meta estimator that employs averaging to increase predictive accuracy and reduce overfitting after fitting numerous decision tree classifiers to diverse subsamples of the dataset. An algorithm for collective machine learning is random forest. Given its good or excellent performance across a wide range of classification and regression predictive modelling problems, it is possibly the most well-known and frequently used machine learning algorithm.

The Gini impurity of random forest classifier is given by:

$$\sum_{i=1}^{C} f_i(1 - f_i)$$

III. RESULTS AND DISCUSSION

A. OVERVIEW

This chapter explains the result of our project and the screenshots for each step are included and explained

B. DATASETS:

The datasets used in our projects are:

DATASETS	CLASSES	EXAMPLES
covid-19	2	13000
Seoul	2	5787
Firing	2	24011

Table 1. Unstructured Dataset

A dataset (typically large collections of files) that isn't stored in a structured database format is referred to as an unstructured dataset.

C. PERFORMANCE METRICS

Our algorithm is evaluated across three metrics: 1) confusion matrix 2) F1- Score 3) Precision 4) Recall. The performance metrics of the collected tweets is compared with eight important supervised machine learning algorithms they are Multinomial Naive Bayes, Bernoulli Naive Bayes, Logistic Regression, Linear SVC, AdaBoost Classifier, Ridge Classifier, Passive Aggressive Classifier and Perceptron.

CONFUSION MATRIX - When the output of a classification problem can be two or more different types of classes, it is the simplest approach to gauge how well the task is performing. A confusion matrix is nothing more than a table containing two dimensions: "Actual" and "Predicted," as well as "True Positives (TP)", "True Negatives (TN)", "False Positives (FP)", and "False Negatives (FN)" in each of the dimensions.

Fig. 11. Confusion Matrix

Calculate the harmonic mean of recall and precision using this score. The weighted average of the precision and recall is the F1 score mathematically speaking. F1 would have a greatest value of 1 and a worst value of 0. F1 score can be calculated using the formula below:

$$F1 = 2 * ((Precision * Recall) / Precision + Recall)$$

The quantity of accurate documents returned by our ML model can be thought of as precision, which is used in document retrievals. By using the confusion matrix and the following formula, we can quickly calculate it:

$$Precision=TP/TP+FP$$

Where,

TP - True Positive FP- False Positive

The quantity of positive results our ML model returned can be referred to as recall. By using the confusion matrix and the following formula, we can quickly calculate it

$$R = TP/TP+FN$$

Classifier	Precision	Recall	F-Score
Multinomial-NB	82.5%	83.2%	84.1%
Bernoulli-NB	87.9%	87.1%	87.9%
Logistic Regression	97.2%	97.4%	97.5%
Linear SVC	98.2%	98.4%	98.1%
AdaBoost Classifier	84.4%	85%	84.6%
Ridge Classifier	86.8%	86.3%	86%
Passive Aggressive	87.6%	87.9%	87.3%
Perceptron	95.5%	95.8%	96.2%
Random Forest Classifier	94.3%	94.6%	95.2%

Table 2. Performance with Evolution metrics

D. RESULTS

Unnamed: 0	date	tweet
0	2022-11-03	@sdqjaan پولیس نے حملہ آور کو ak-47 کے ساتھ کی...
1	2022-11-03	black day for pakistan 😭 😭 #firing #imrankhan #ع...
2	2022-11-03	imran khan himself has created this drama to e...
3	2022-11-03	shooter details* name: general qamar javed b...
4	2022-11-03	only army bajwa's bastards are behind him. fuc...
...	...	...
24007	2022-11-03	wondering about firing incident on pti dharna ...
24008	2022-11-03	will rana sanaullah be accountable for what he...
24009	2022-11-03	@gfarooqi abay aur firing papian dene ke liye ...
24010	2022-11-03	لانگ مارچ میں بہت سی انسانی جانوں کا نقصان ہ...
24011	2022-11-03	saw the devastated news!!! i hope he's safe an...

Fig. 12. Data Extraction using Twint

Above figure shows the data extracted using Twint. In which 24,011 tweets has been scrapped from the twitter.

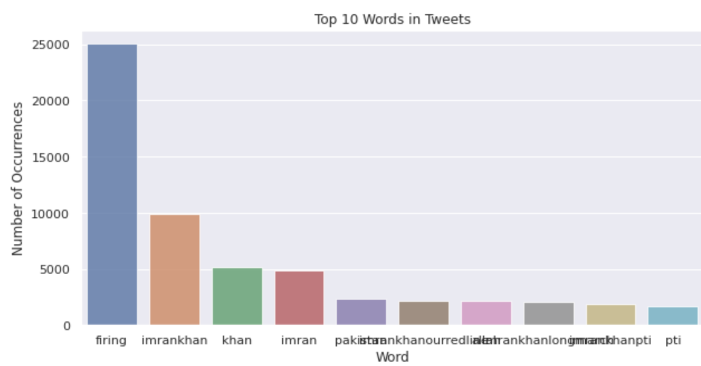

Fig. 13. Unigram Analysis

Above graph shows the top 10 words in the extracted tweets and how much times the word occurred in the overall tweets.

Fig. 14. Word Cloud

Above figure shows the word cloud for what are the keywords repeatedly used by the users

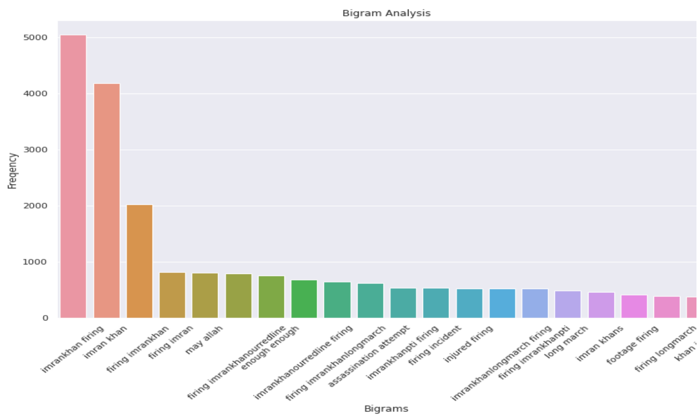

Fig.15. Bigram Analysis

Above graph shows the bigram analysis in which top 20 words are visualizedand the occurrences of the words

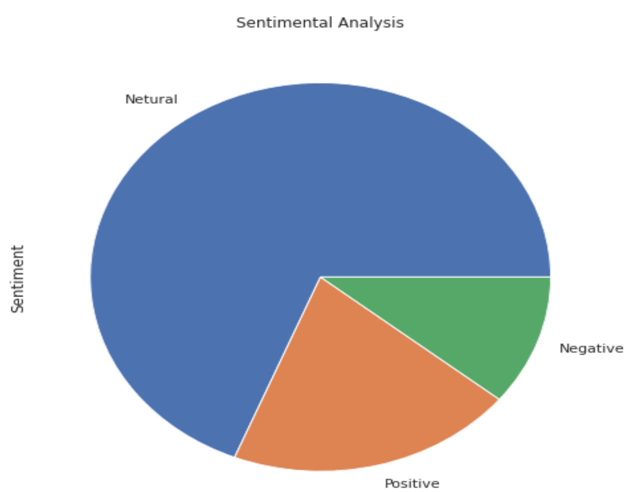

Fig. 16. Sentiment Analysis Pie Chart

Above graph shows the sentiment analysis pie chart for Sentiments VS Number of Tweets and classify the tweets into positive, negative and neutral

DATA SETS	PURITY	F MEASURE
Covid19	0.88	0.86
Seoul	0.88	0.85
Firing	0.95	0.96

Table 3. Performance of unstructured datasets

Classifiers	Accuracy (%)
MutinomialNB	89.6
BernouliNB	85.5
Logistic Regression	85
Linear SVC	**98.4**
Ada Boost	90.9
Ridge Classifier	97.5
Passive Aggressive	98.2
Perceptron	98
Random Forest Classifier	96.5

Table 4. Performance of the classifiers

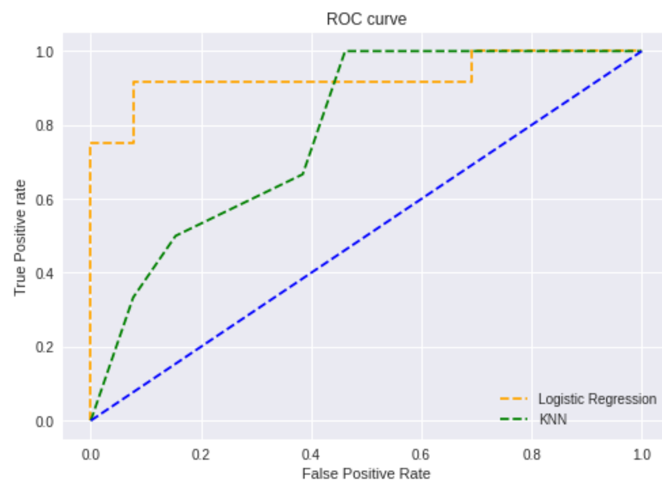

Fig. 17. ROC Curve for true positive vs false positive rate

VI. CONCLUSION

The number of people using social media every day is dramatically rising. Social media is the preferred platform for people to express their genuine ideas over face-to-face interactions. We looked at the general public's overall response to how various residents implemented trending tweets using posts from Twitter. After annotation and preprocessing, we used eight supervised machine learning approaches with various types of text. The LinearSVC classifier and unigram exhibit the best performance, according to our observations. The combination offers us a 98.4% accuracy rate, which is the highest of all the combinations we tested using our data set. We combined the performance by calculating accuracy, recall, F1-Score, and tenfold cross-validation for all the combinations, and we found that LinearSVC and unigram produced the best results. The public's tweets on trending topics were then subjected to sentiment analysis using this combination, and we discovered that nearly half of the population (48.69%) is speaking neutrally about the topics, 29.81% are speaking positively, and 21.5% are feeling negatively for some reason.

V. FUTURE WORK

We intended to implement deep learning models for future work that would forecast the sentiment of tweets and categorise each other's tweets as positive, neutral, or negative. By utilizing Convoluted neural network model, we will be able to achieve a more stable and accurate predictive modal.

VI. REFERENCES

[1] J. T. Wu, K. Leung, and G. M. Leung, "Nowcasting and forecasting the potential domestic and international spread of the 2019-nCoV outbreak originating in Wuhan, China: A modeling study," Obstetrical Gynecolo. Surv., vol. 75, no. 7, pp. 399–400, Jul. 2020.

[2] S. Li, Y. Wang, J. Xue, N. Zhao, and T. Zhu, "The impact of COVID-19 epidemic declaration on psychological consequences: A study on active Weibo users," Int. J. Environ. Res. Public Health, vol. 17, no. 6, p. 2032, Mar. 2020.

[3] WHO Statement Regarding Cluster of Pneumonia Cases, WHO, Wuhan, China, 2020.

[4] R. Pandey et al., "A machine learning application for raising WASH awareness in the times of COVID-19 pandemic," 2020, arXiv:2003.07074. [Online]. Available: http://arxiv.org/abs/2003.07074

[5] A. S. M. Kayes, M. S. Islam, P. A. Watters, A. Ng, and H. Kayesh, "Automated measurement of attitudes towards social distancing using social media: A COVID-19 case study," Tech. Rep., Oct. 2020.

[6] C. K. Pastor, "Sentiment analysis on synchronous online delivery of instruction due to extreme community quarantine in the Philippines caused by Covid-19 pandemic," Asian J. Multidisciplinary Stud., vol. 3, no. 1, pp. 1–6, Mar. 2020.

[7] A. D. Dubey, "Decoding the Twitter sentiments towards the leadership in the times of COVID-19: A case of USA and india," SSRN Electron. J., Apr. 2009, doi: 10.2139/ssrn.3588623.

[8] L. Chen, H. Lyu, T. Yang, Y. Wang, and J. Luo, "In the eyes of the beholder: Analyzing social media use of neutral and controversial terms for COVID-19," 2020, arXiv:2004.10225. [Online]. Available: http://arxiv.org/abs/2004.10225

[9] G. Barkur, Vibha, and G. B. Kamath, "Sentiment analysis of nationwide lockdown due to COVID 19 outbreak: Evidence from India," Asian J. Psychiatry, vol. 51, Jun. 2020, Art. no. 102089.

[10] M. Alhajji, K. A. Al, M. Aljubran, and M. Alkhalifah, "Sentiment analysis of tweets in Saudi Arabia regarding governmental preventive measures to contain COVID-19," Dept. Social Behav. Sci., College Public Health, Temple Univ., Philadelphia, PA, USA, Tech. Rep., doi: 10.20944/preprints202004.0031.v1.

[11] J. Samuel, A. GG, M. Rahman, E. Esawi, and Y. Samuel, "Covid-19 public sentiment insights and machine learning for tweets clas- sification. Nawaz and Rahman, Md. Mokhlesur and Esawi, Ek and Samuel, Yana," Information, vol. 11, no. 6, pp. 1–22, Apr. 2020, doi: 10.3390/info11060314.

[12] R. Liu, Y. Shi, C. Jia, and M. Jia, "A survey of sentiment analysis based on transfer learning," IEEE Access, vol. 7, pp. 85401–85412, 2019.

[13] N. Kaka et al., "Digital India: Technology to transform a connected nation," McKinsey Global Inst., India, Tech. Rep., Mar.2019.[Online].Available:https://www.mckinsey.com/~/media/McKinsey/Business%20Functions/McKinsey%20Digital/Our%20Insights/Digital%20India%20Technology%20to%20transform%20a%20connected%20nation/MGI-Digital-India-Report-April-2019.pdf

[14] A. Abd-Alrazaq, D. Alhuwail, M. Househ, M. Hamdi, and Z. Shah, "Top concerns of tweeters during the COVID-19 pandemic: Infoveillance study," J. Med. Internet Res., vol. 22, no. 4, Apr. 2020, Art. no. e19016.

[15] P. Burnap et al., "Tweeting the terror: Modelling the social media reaction to the woolwich terrorist attack," Social Netw. Anal. Mining, vol. 4, no. 1, p. 206, Dec. 2014.