Sentiment Analysis of 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 Trending 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 - Include a list of important index terms here. Machine learning, Imran Khan, NLP, and firing.

I. INTRODUCTION

A. **SENTIMENT ANALYSIS**

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.

B. **NATURAL LANGUAGE PROCESSING**

Natural language processing (NLP) defines the branch of computer technology—and greater in particular, 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.

C. **TEXTBLOB**

TextBlob defined as Lexicon-based sentiment instrument with some predefined rules, wherever it's some scores that facilitate to calculate a sentence's polarity. that is why the Lexicon-based sentiment analyzers also are referred to as "Rule-based sentiment analyzers". TextBlob could be a Python library for process matter knowledge. It provides a straightforward API for diving into common language process (NLP) tasks like part-of-speech tagging, phrase extraction, sentiment analysis, classification, translation, and more.

D. **VADER**

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 optimised 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.

II. LITERATURE STUDY

A OVERVIEW:

To analyze the sentiment of trending tweets posted by different citizens whether the data is positive ,negative or neutral tweets. To Handle Sentiment Analysis using NLP and Machine Learning Algorithms.

B DATA EXTRACTION [1]:

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".

C DATA LABELING [2]:

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 TextBlob and VADER results to consolidate the polarities.

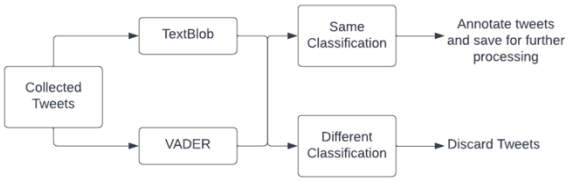

Fig. 1. Data Labeling

D DATA PREPROCESSING [3]:

The data we've 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 implemented 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.

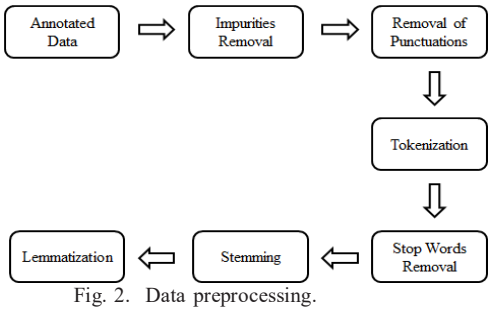

Fig. 2. Data preprocessing.

E VECTORIZATION [4]:

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 CountVectorizer feature extractor to calculate word frequencies. CountVectorizer 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 approach associate actual juvenile person is young." CountVectorizer 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 COUNTVECTORIZER

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

Fig. 3. CountVectorization

F TRAINING AND TESTING [5]:

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 NaiveBayes, Bernoulli NaiveBayes, LogisticRegression, LinearSVC, AdaBoostClassifier, RidgeClassifier, PassiveAggressiveClassifier, 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.

III. SYSTEM STUDY

A SCOPE:

The scope of Twitter Sentiment Analysis for trending topics is to analyze the tweets posted by different peoples sentiments and opinions.

B PRODUCT FUNCTION:

We have used Twint, Open source package for extracting data for the keyword "Firing" nearly 24,011 tweets have been scrapped using tweepy package.

We have labelled the data using TextBlob and VADER which is used to classify the tweets as Positive, Negative or Neutral tweets.

Once, We have labelled the tweets using TextBlob and VADER we plotted graph and counted how much Positve, Negative or Netural tweets are present.

After finishing the data labelling, the data we have collected may hold some unsought and sentiment fewer words like links, Twitter-specific words such as 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 training and testing. To amend classifier efficiency, it is necessary to remove noise from the labeled data set before feeding the classifier.

We separate noise from labeled data in our pre-processing module.

We implemented a module to remove the above-specified impurities, converted the data set into a data frame, and then removed string punctuation, tokenized, and removed English stop words, stemming, and lemmatized the data.

We have used the CountVectorizer feature extractor to calculate word frequencies.CountVectorizer counts the frequency of each word present in the document and creates a sparse matrix.The matrix is not sparse because we are converting the only single document.

In the case of multiple documents, it is frequent that a word present in one document can be missing from some other documents, and hence the corresponding cells are filled up with zero, and the resultant matrix will become sparse.

After feature extraction of the preprocessed data set, we have passed the data to machine learning classifiersWe have used eight classifiers (Multinomial NaiveBayes,Bernoulli NaiveBayes,LogisticRegression,LinearSVC,AdaBoostClassifier,RidgeClassifier,PassiveAggressiveClassifier andPerceptron) for this purpose.

We have used 80% data for training and 20% data for testing the classifiers.

IV. SYSTEM DESIGN

4.1 OVERVIEW

This section presents the overview of the whole system.

4.2 OVERALL ARCHITECTURE

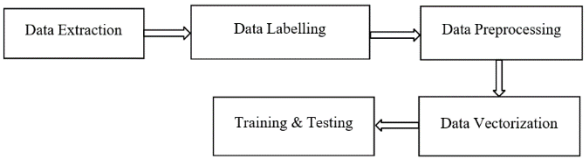

Figure 4.2.1 System architecture

4.3 MODULES

- •Multinomial NaiveBayes,
- •Bernoulli NaiveBayes
- •LogisticRegression
- •LinearSVC
- •AdaBoostClassifier
- •RidgeClassifier
- •PassiveAggressiveClassifier
- •Perceptron
- •Random Forest Classifier

4.3.1 MULTINOMIAL NAIVEBAYES:

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.There is also the possibility of using fractional counts such as tf-idf in practice.

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

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

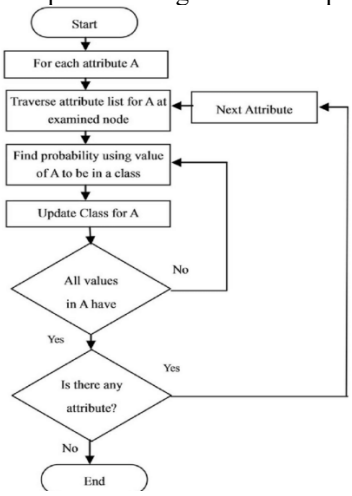

Figure 4.3.1.1 Work Flow for Multinomial NaiveBayes

4.3.2 BERNOULLI NAIVEBAYES:

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.'

$$q = 1 - (\text{Success probability})$$
$$q = 1 - p$$

$$p(x) = P[X = x] = \{ \blacksquare(q = 1 - p \quad x = 0 @ p \quad x = 1)\dashv$$

$$x = \{ \blacksquare(1, \quad \& Bernoulli\ Trail = S @ 0, \quad \& Bernoulli\ Trail = F)\dashv$$

4.3.3 LOGISTIC REGRESSION:

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 amount 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:

$$y/(1-y);0 \text{ for y=0 and infinity for y=1}$$

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

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

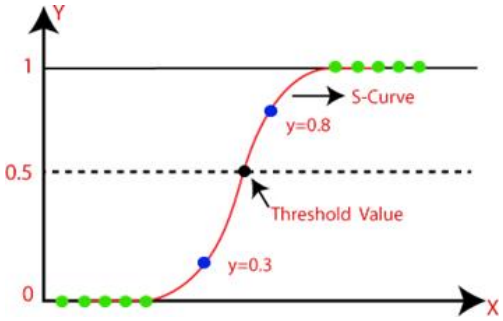

Figure 4.3.3.1 Sigmoid Function in Logistic Regression

4.3.4 LINEAR SVC:

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. An 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.

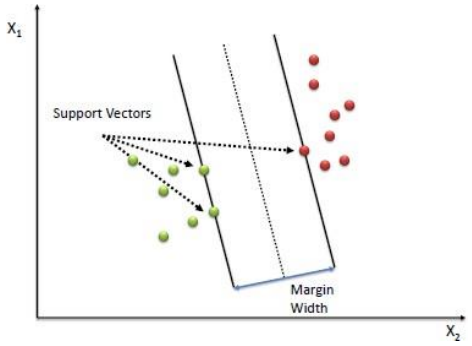

Figure 4.3.4.1 Support Vectors and Margin Width

4.3.5 ADABOOST CLASSIFIER:

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.

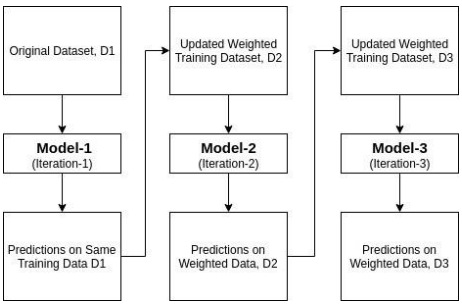

Fig 4.3.5.1 Work Flow of AdaBoost Classifier

$$\alpha = 1/2 * \ln(\ (1 — E)/E)$$

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

4.3.6 RIDGE CLASSIFIER:

A Ridge regressor is essentially a Linear Regressor that has been regularised. In other words, we add a regularised 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 regularised term's 'alpha'

parameter regulates the model's regularisation, 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 loss function and L2 regularisation (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.

4.3.7 PASSIVE AGGRESSIVE CLASSIFIER:

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

$$\begin{cases} X = \{\bar{x}_0, \bar{x}_1, \dots, \bar{x}_t, \dots\} \, where \, \bar{x}_i \in \mathbb{R}^n \\ Y = \{y_0, y_1, \dots, y_t, \dots\} \, where \, y_i \in \{-1, +1\} \end{cases}$$

The prediction is easily determined as follows given a weight vector w:

$$\tilde{y}_t = sign(\bar{w}^T \cdot \bar{x}_t)$$

4.3.8 PERCEPTRON:

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.

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

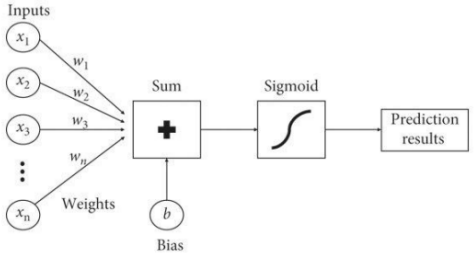

Figure 4.3.8.1 Work Flow for Perceptron

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 perceptrons. Non-linear input vectors are difficult to properly categorize.

4.3.9. RANDOM FOREST CLASSIFIER:

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:

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

V. IMPLEMENTATION METHODOLOGY

5.1 OVERVIEW:

The implementation methodology describes the main functional requirements, which needed for doing the project.

5.2 ESSENTIAL LIBRARIES:

The library used in this project are pandas, textblob, nltk, countvectorizer,matplotlib.

5.2.1 PANDAS:

The Python library Pandas offers quick, adaptable, and expressive data structures that are intended to make working with "relational" or "labelled" data simple and natural. It aspires to serve as Python's core, high-level building block for performing useful, in-the-real world data analysis.

5.2.2 TEXTBLOB:

A Python (2 and 3) package called TextBlob 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.

5.2.2.1 NLTK:

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.

5.2.3 COUNT VECTORIZER:

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

5.2.4 MATPLOTLIB:

One of the most common programmes for Python statistics visualisation is Matplotlib. 2-Dimensional plots can be created from records in arrays using this cross-platform library. Python is the language that was used to create Matplotlib. As an open source option, Matplotlib with NumPy is available.

5.2.4.1 MATPLOTLIB PYPLOT:

The matplotlib is the command-style utilities in pyplot enable matplotlib to operate similarly to MATLAB. Each pyplot feature makes some interchange to a determine, such as creating a figure, a plotting region in a determine, charting some traces in a plotting vicinity, embellishing the plot with labels, etc. matplotlib was used for this task. Pyplot is used to create graphs, such as bar graphs and Cartesian graphs, showing the outstanding Parameters Vs overall performance of the programme.

5.3 FUNCTIONS USED FOR IMPLEMENTATION:

The user defined function used for the implementation of the project are

5.3.1 CLEAN TWEET

For the purpose of removing any noisy data from the tweets we collected, we employed the clean tweet approach. By removing links, @mentions, RTs, punctuation, and other elements, a structured tweet with useful information can be produced.

5.3.2 POLARITY:

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.

5.3.3 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.

5.3.4 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.

5.3.5 LEMATIZATION:

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.

PERFORMANCE METRICS

6.1 OVERVIEW:

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 witheight 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.

6.2 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.

6.3 F1-SCORE:

We can 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:

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

6.4 PRECISION:

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

6.5 RECALL:

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.

$$R = TP/TP+FN$$

RESULTS AND DISCUSSION

7.1 OVERVIEW

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

7.2 DATASETS:

The datasets used in our projects are:

7.2.1 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.

DATASETS	CLASSES	EXAMPLES
covid19	2	13000
seoul	2	5787
firing	2	24011

Table 7.2.1 Unstructured Dataset

7.3 SCREENSHOTS

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...

Figure 7.3.1 Data Extraction using Twint
Above figure shows the data extracted using Twint. In which 24,011 tweets has been scrapped from the twitter.

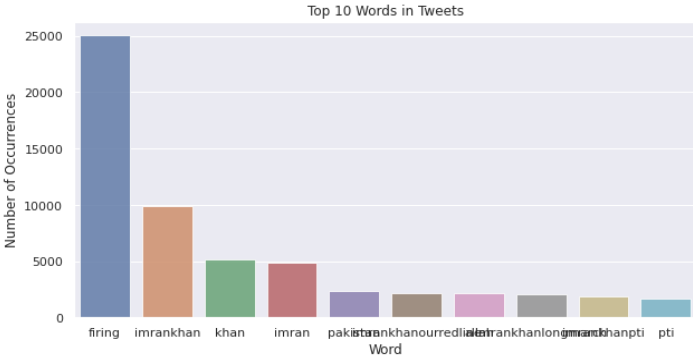

Figure 7.3.2 Unigram Analysis
Above graph shows the top 10 words in the extracted tweets and how much times the word occurred in the overall tweets.

Figure 7.3.3 Word Cloud
Above figure shows the word cloud for what are the keywords repeatedly used by the users

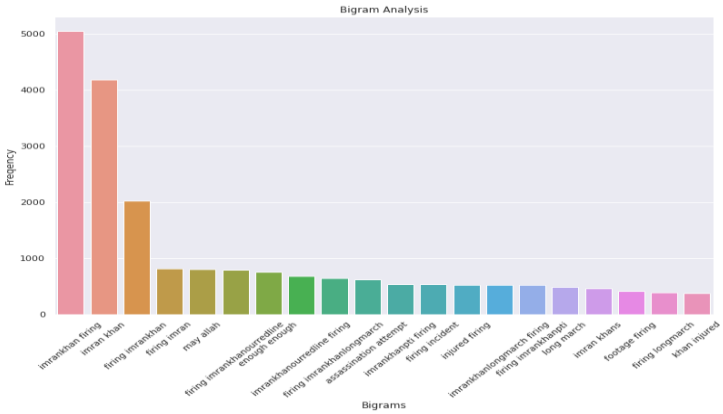

Figure 7.3.4 Bigram Analysis

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

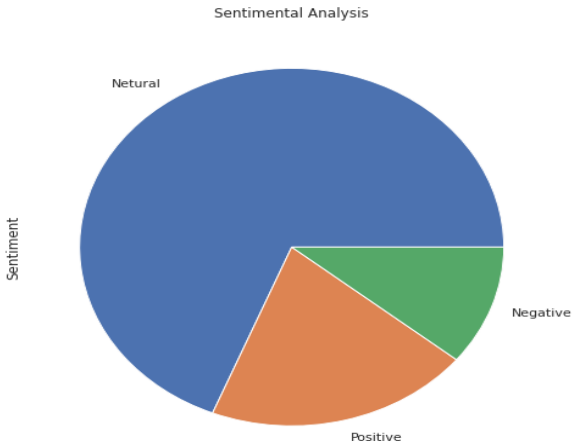

Figure 7.3.5 Sentiment Analysis Pie Chart
Above graph shows the sentiment analysis pie chart for Sentiments VS Number of Tweets and classify the tweets into positive, negative and neutral

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

Table 7.3.1 Performance of unstructured dataset

Classifiers	Accuracy (%)	Rank
MutinomialNB	89.6	7
BernouliNB	85.5	8
Logistic Regression	85	9
Linear SVC	98.4	1
Ada Boost	90.9	6
Ridge Classifier	97.5	4
Passive Aggressive	98.2	2
Perceptron	98	3
RandomForest Classifier	96.5	5

Table 7.3.2 Performance of the classifiers

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%
RandomForest Classifier	94.3%	94.6%	95.2%

Table 7.3.3 Performance with Evolution metrics

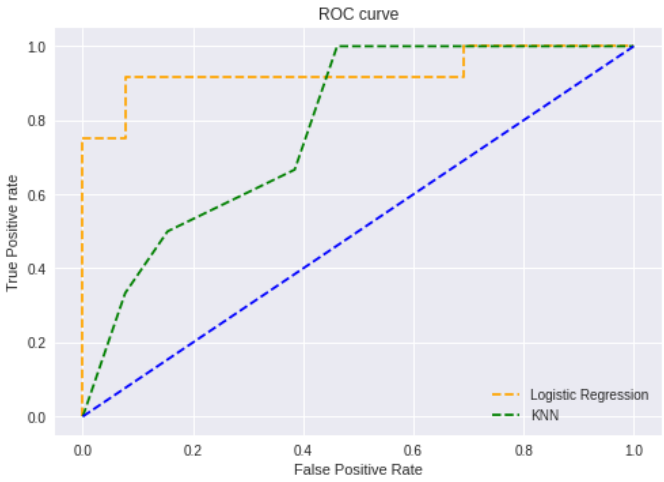

Figure 7.3.5 ROC Curve for true positive vs false positive rate

CONCLUSION

8.1 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.

8.2 FUTURE WORK

We intended to create a web-based tool for future work that would forecast the sentiment of tweets and categorise each other's tweets as positive, neutral, or negative.

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.