# AN INTELLIGENT SENTIMENT ANALYSIS SYSTEM BASED ON RECENT TRENDS IN MACHINE LEARNING

**Khan Awais Khan*, Saadat Hanif Dar, Samina Fazilat, Muhammad Raees, Sehrish Manzoor, Muhammad Usama Ayub, Afzal Ahmed, Areeb Ahmed Mir**

Department of Software Engineering, Mirpur University of Science and Technology, MUST-Mirpur AJK, Pakistan 10250.

*Email: saadat.dar@gmail.com

***ABSTRACT-*** *Social networks are producing data at an ever-growing rate. Social nets are the platforms for conveying opinions (or sentiments) on varying concerns. Sentiment analysis is a way of determining the subjectivity of the message/opinion. The sentiment analysis on social nets (like Twitter), assesses the sentiments represented by the users in their tweets with tweets expressing ideas, interests, and opinions in a range of settings. Consequently, Twitter sentiment analysis is a hot research area at present. We aim to concentrate on sentiment analysis on Twitter data, which is beneficial for discovering insight into the words used in the tweets, where opinions are strong, unstructured, and have either positive, negative, or neutral influence. Tweets are an amalgam of complex characters and complicated expressions requiring efficacious computation. In this study, we present various algorithms and methodologies employed on Twitter data for sentiment analysis. Here, the goal of sentiment analysis is to gauge or quantify the emotional perspective of the author against a certain issue.*

**Keywords:** Sentiment Analysis, Text Classification, Machine Learning.

## I. INTRODUCTION

Nowadays, social media is omnipresent, and everyone may have their views represented on it. During the month of December 2018, 1.52 billion Facebook users were active diurnally [1]. People are spending more time on social media; about three out of every four adult Internet users now utilize at least one social media site [2]. People of all ages and backgrounds use this technology to express themselves. The universal reach of social networks enables them to be powerful carriers of expressions, such as product reviews, political statements, or reactions. Twitter is one of the most popular micro-blogging services among many social networks [3]. Twitter is the go-to medium for product reviews, and businesses utilize it as a means of measuring the effectiveness of their offerings. The information contained in Tweets (i.e., sentiments) is critical to any company aspiring to enhance sales, growth, or attempt to turn around a failing product. However, Twitter data is massive and requires expedient handling. Machine learning and data mining techniques are beneficial to deal with a large number of tweets in fewer increments, which provides effective results and useful data [4]. Many real-world applications employ micro-blog data to analyze and to conclude world trends, events, or incidents [5].

Social networks advanced dramatically due to the rapid digitization of the service industry and innovative information technology impacting hundreds of millions of users [6, 7]. Opinion mining and sentiment analysis revolve around associated concepts including sentiments, evaluations, attitudes, and emotions. Sentiment analysis has emerged as one of the most important fields of research in natural language processing since the early 2000s [8]. Several tools to classify sentiments about a product or service are widely available, companies have challenges when extracting data. Extracting, evaluating, and arranging tweets in an acceptable format is challenging. Informal language can hinder the process of detecting sentiment in some systems. Here we perform sentiment analysis to detect user interests using a semantic framework, important keywords, and opinion words from tweets and then provide their polarity. The objective of this research is to examine how sentiment analysis could be applied to learning analytics. Additionally, to strengthen research applicability we have concentrated on the different sentiment analysis approaches and models to evaluate results. The contributions of this research study are three-fold as under.

1) The extraction and preparation of data acquired from Twitter using third-party libraries and generating their polarities into negative, positive, and neutral attitudes.
2) Exploratory Data Analysis (EDA) is performed on unstructured textual data (tweets) to elicit insights about polarity generation and opinion subjectivity in tweets.
3) The performance of various machine learning models is compared to establish the applicability of sentiment analysis methods on Twitter data. The competitive analysis and performance show the decision tree algorithm outperforms other models. The approach will also help to examine and set the benchmark for prospective applications in Twitter sentiment analysis.

A concise introduction and oversight of the proposed work is elicited in this section. Some discussion about sentiment analysis, pre-processing, and use of machine learning methods is described in Section 2. Implementation of pre-processing on the dataset and model's execution is provided in section 3. Section 4 presents an evaluation of experiments and results. Finally, we conclude our paper in section 5.

## II. RELATED WORK

Sentiment analysis has been addressed at numerous degrees of granularity as a Natural Language Processing task. The Sentiment Classification framework has been built using multiple feature sets and classification approaches that have been combined into one system [9]. These sets include Naive Bayes, Maximum Entropy, and Support Vector Machines as basic classifiers. A study for classifying Arabic tweets included many subtasks, one of which being TFIDF and Arabic stemming [10, 11]. A study [12] analyzed social issues' sentiments and opinions through literary reviews. Classification approaches of several sorts when combined produced superior results according to authors [13]. One sentiment analysis study [14] gathered data from Global Support Services feedback survey data. The primary goal of their research is to investigate the aspects of language that include feature tags like POS. The experimental results show that using linguistic feature analysis contributes to classifier accuracy [15]. These researchers used hashtags to do graph-based categorization and an algorithm for emoticon weighting [16]. Classifier accuracy can be acquired by a combination of methods, including feature selection, data testing, and demonstrating the data's ability to express abstract language features [17]. In literature, some methods analyze how information could be derived from analyzing other categories and associated patterns [6]. The recognition method [18] followed by statistics-based techniques to find sentiments connected to a subject, can use a Markov model-based tagger. A study [19] took Twitter's data to see if it could successfully forecast an election. To explore if Twitter is being utilized as a platform for political discourse, they had to study the setting of the German federal election [20]. A psychometrically verified lexicon is used in text analysis software that identifies and rates emotional, cognitive, and structural elements in text samples. Political sentiments were extracted from this data using 12 variables, including positive and negative emotion, grief, worry, rage, certainty, tentativeness, achievement, and money [21, 22]. Moreover, natural language processing techniques are used to identify sentiment linked to the issue. Most educational institutions monitor teaching and learning processes, teaching materials, and evaluations to improve the quality of education they deliver. When feedback results are used in the learning process, it can aid in enhancing the overall quality of the process, and therefore better learning outcomes are achievable. [24]

## III. METHOD AND MODEL DESIGN

The method employs an extracting technique, followed by pre-processing from tweets and then classifying tweets based on their semantics. To avoid information loss, an enhancer of knowledge is used to enhance the process of knowledge removal from the tweets obtained. Analysis of sentiments demonstrates people's approach to many subjects. These data can also contribute to generating a detailed user profile and relevant recommendations. The detailed process of the methodology being employed is depicted in the following figure.

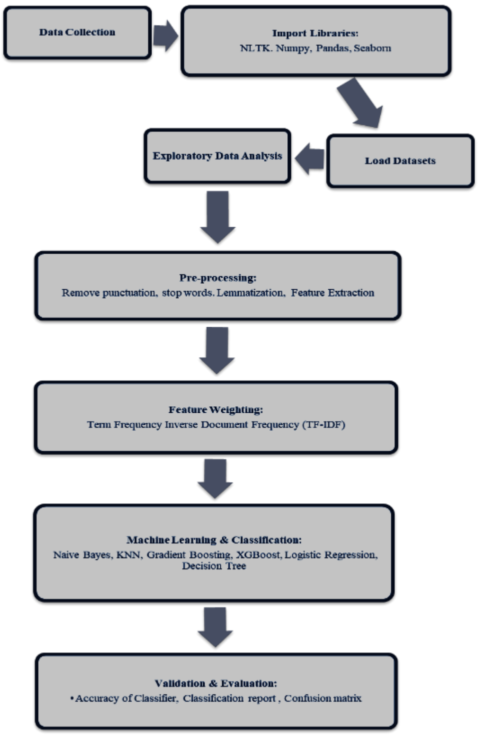

Data Collection
Import Libraries:
NLTK. Numpy, Pandas, Seaborn
Load Datasets
Exploratory Data Analysis
Pre-processing:
Remove punctuation, stop words. Lemmatization, Feature Extraction
Feature Weighting:
Term Frequency Inverse Document Frequency (TF-IDF)
Machine Learning & Classification:
Naive Bayes, KNN, Gradient Boosting, XGBoost, Logistic Regression, Decision Tree
Validation & Evaluation:
• Accuracy of Classifier, Classification report , Confusion matrix

**Figure 1 Flow of Methodology**

For data collection, we chose Twitter as it is quite popular these days. Hashtags or keywords can be used to pull tweets from Twitter using an API. One of the limits imposed by Twitter is that the Twitter Search API can only produce a limited number of tweets at a time. Tweepy can be used to access the Twitter API through Python. The tool is designed to deal with authentication, connectivity, and many other related services. The following diagram shows the data collection process.

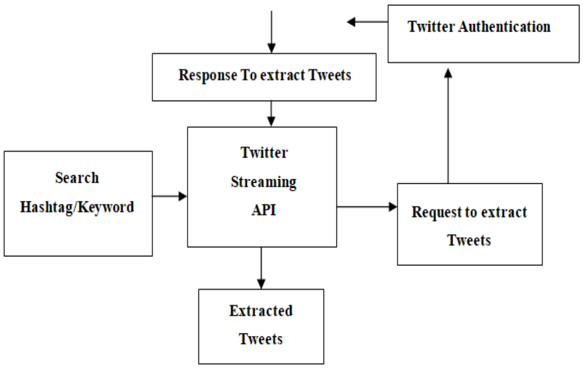


**Figure 2. Data Collection Process**

### 3.1 Pre-processing

Applying text pre-processing (e.g., POS, eliminating URLs, extending acronyms, substituting negative statements, stemming, removing stop words) is the most common technique used to determine sentiment polarity in tweets. To obtain a more effective classifier, it is believed that pre-processing of the data helps to reduce the noise in the text. Results show that sentiment classification accuracy may be improved greatly with pre-processing with relevant features and a representation [24]. Technically accurate, though, raw tweets pulled from Twitter tend to yield a noisy and poorly understood dataset. This is due to how people use social media casually and ingeniously. Tweets exhibit some distinctive properties, such as retweeting, emoticons, user mentions, etc., and these features should be obtained for the analysis. Thus, to produce a dataset that is suitable for classification, raw Twitter data must be normalized. To clean the dataset and reduce its size, we have implemented numerous pre-processing methods. Preparation of a dataset for processing includes the removal of all extraneous data, such as HTML tags, white spaces, and characters with a specific purpose. The source of this noise is nonsensical, and hence must be eradicated. A regular expression (RE) Python library data cleaning is completed using that method. For our system, the cleaning process is as follows:

i. URL removal because of their simplicity, not their essentiality.
ii. Users such as @abs, which add no weights to sentiment classification, are also deleted.
iii. Delete the punctuation. Then, special character removal happens.
iv. Finally, white space is replaced in multiple places across the document.

Stop words are a word bag based on a dictionary. These are the usual words not only in English but in any other language. Stop words focus primarily on significant terms rather than commonly used words in a language. The removal of unnecessary terms from the Twitter data set is carried out so that the resulting data set has only the information needed for the study. Only important words that could lead to feeling detection are left after word elimination. Stop word removal and tokenization by another python library called NLTK.

Lemmatization is to reduce inflectional forms and sometimes derivative forms of a word to a single form of basis. Lemmatization usually concerns the right use of a word vocabulary and a morphological analysis, which normally just seeks to remove inflection marks and return to the base or dictionary form of a word recognized as a lemma. Whereas Stemming refers to a ground heuristic strategy that usually cuts out the endings of words in the expectation of this aim and often entails the removal of derivative appeals. Feature extraction is an important step in generating a list of objects, aspects, characteristics, and opinions. It is intended to extract sentences containing one or more characteristics, aspects, and opinions. In most circumstances, words in aspect are substantives and substantive phrases, and their opinions are adjectives and adverbs. Only necessary words are left in tweets for analysis following the pre-processing phase. We only extract tweets with nouns and sentences. These substantive terms are used to know who is in the tweet. Only words that include characteristics or aspects like adjectives and adverbs are left after extracting nouns and sentences. Therefore, these retrieved features are sorted into feelings in the next phase. Features must represent the data information in a manner that best matches the problem resolution method. Although some intrinsic features can be recovered directly from raw data, we usually need derived features that are relevant to the underlying problem.

TF-IDF is one of the statistical measures and a loading technique. Common verbs, adjectives, and substances are retrieved from the processed dataset used to calculate the sentiment polarity, for example positive, negative, and neutral in a sentence. This allows replicas such as unigrams, bigrams, or n-grams to find out people's opinions on the target topic. To calculate the importance of a word in a document we use the TF-IDF method, in which weight is indicated for each word in the document. It extracts characteristics based on the count of words, giving a few frequent words less weight and a rarer weight.

## 3.2 Modeling and Classification

The Tweet Sentiment Analysis can be collecting relevant information from two fundamental methodologies, one lexicon-based and the other machine learning approach. Machine learning strategies are used to train the algorithms to incorporate the train data into the test data with the algorithm. Different types of classifiers are used for the construction of machine learning models, including the Multinomial Naive Bayes, KNN, Gradient Boosting, XGBoost, Logistic Regression, and Decision Tree.

### 3.2.1 Naive Bayes

Naive Bayes is a supervised machine learning technique used largely for classification. In this case, "supervised" means that the algorithm was taught given both inputs and category outputs (In other words, the data has the proper desired outcome at each point, which the algorithm should predict.). But why is this algorithm referred to as "naive"? Because the classifier anticipates that the model's input features will be mutually independent. As a result, changing one input function has no influence on the others. As a result, it is naive in the sense that this hypothesis could be true or false. Naive Bayes has several advantages, one of which is that it uses a probabilistic method, all computations are performed in real-time and on the fly.

### 3.2.2 KNN:

K-Nearest Neighbour termed KNN is a supervised learning method for both regression and classification tasks. It is commonly used in machine learning for categorization challenges. KNN operates on the notion that all data points falling close to one another fall into the same class. In other words, new data points groups with similarities are classified. The KNN algorithm retains all data available and identifies a new data point depending on the measure of similarity (e.g., distance functions). This means when fresh information is displayed. Then the K-NN method can be simply assigned to a suitable category.

### 3.2.3 Gradient Boosting:

The main distinction between AdaBoost and the Gradient Booster is that GBMs employ a different way to calculate whether students misidentify data points. AdaBoost determines where a model is inadequate by analyzing heavily weighted data points. In the meantime, GBMs use gradients to assess the precision of students by implementing a loss function to a model. Loss functions are a means to measure the precision of a model fit into the dataset, calculate an error, and optimize the model to minimize that error. GBMs allows the user to optimize a given loss function in accordance with their intended objective.

A gradient boosting machine or a GBM combines many decision-making predictions to produce the final predictions. Keep in mind that decision trees are all the weak students in a gradient booster. However, if we use the same technique, how can we better use 100 decision trees than one decision tree? How do different decision-making trees capture different signals/data information? Here is the trick: the nodes choose another subset of features in each decision tree to choose the best split. This means that individual trees are not all identical and might therefore collect diverse indications from the data. Furthermore, each new tree takes the mistakes and errors of the preceding trees into consideration. Thus, on the faults of prior trees, every succeeding decision tree is formed. That's how the trees are formed sequentially in a gradient boost method.

### 3.2.4 XGBoost:

XGBoost is a group of decision-making bodies. These trees are individually bad models, but when clustered together they can be incredibly effective. The distinction between XGBoost and Random Forest is the construction and combination of the trees. Random Forest constructs fully cultivated decision-making bodies on data sub-samples in simultaneously. Each tree is very specialized for predicting and not generalizing well on its subsample (high variance). The Random Forest technique lowers variation and provides superior performance by integrating the forecasts of each individual tree.

On the other hand, XGBoost makes iteratively short and basic decision trees. For their significant prejudice, each tree is called a "weak student." XGBoost begins by building a first basic tree with poor performance. It then builds another tree that is trained to predict what the original tree couldn't do and is a weak student. The algorithm continues to generate more weak learners sequentially, and each will correct the preceding tree until a stop is reached as is the number of trees to be constructed (estimators).

### 3.2.5 Decision Tree:

Decision trees are used to validate parametric categorization and regression. It is a classifier that uses data-collecting attributes, rules for decisions, and results. A decision tree has two nodes: a node for deciding and a node for discovering a possible outcome. Decision nodes are responsible for decisions, while leaf nodes do not bear any other branches. While it is the dataset that determines the decisions and tests that are made with it, the dataset also influences how decisions and tests are formed. When thinking about a decision, a solution/decision tree is a visual representation of all the many options. In the decision tree, the root node resembles a tree, and from there, multiple branches branch out, forming a tree-like structure. A model should be developed to calculate the value of a target variable based on information about data elements that dictate simple decision-making principles. One can think of a tree as made up of several different sections. Decision tree models can be developed using tree structure classification or regression trees. As the amount of data is progressively decreased, a decision tree is built on the fewer and smaller portions of data. Each individual leaf is connected to an individual decision node, as well as all the other leaves on the tree. A decision node may have numerous possible avenues of action. To state an opinion, you must establish a value in your statement, using a leaf node. a root node is the strongest predictor Categorical and numerical data can both be tackled with a decision tree.

## IV. RESULTS

We have around 10000 tweets in the training and test set. We use a symbolic and quantitative natural language processing library known as the Natural Language Toolkit (or NLTK). NLTK is intended to assist researchers and educators in natural language processing (NLP) and closely related subjects such as analytical linguistics, social psychology, machine learning, retrieval systems, and machine learning. In addition to being utilized as a teaching tool and an individualized learning tool, NLTK has also been effectively used as a forum for researchers to prototype and create research systems. Features for grouping, tokenization, stemming, tagging, parsing, and semantic reasoning are all supported by the NLTK. We performed an exploratory data analysis first. The technique of data analysis is crucial when it comes to data science. It allows the data scientist to make sense of the data that is provided. It is crucial to understand the usage of EDA to support findings and to evaluate the trustworthiness of data. The following diagrams result of EDA to get insights into the dataset used in the study. The kernel distribution of words is calculated based on the words inside each tweet and plotting the graph. The difference in the number of words and Jaccard scores among distinct Sentiments are fascinating to see. For the same reason, KDE of Jaccard scores of positive, negative, and neutral tweets, a distribution plotted.

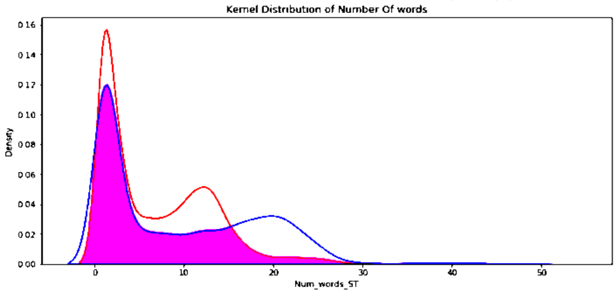


**Figure 3. Kernel Distribution of Number of Words**

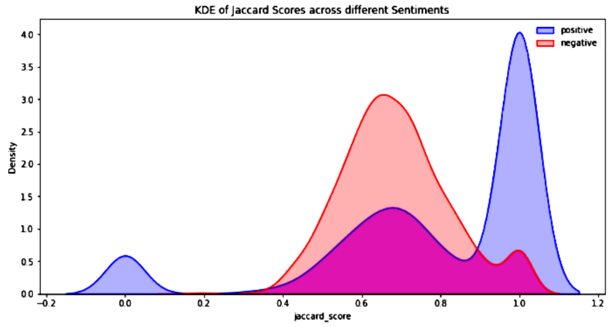


**Figure 4. KDE of Jaccard scores across different sentiments**

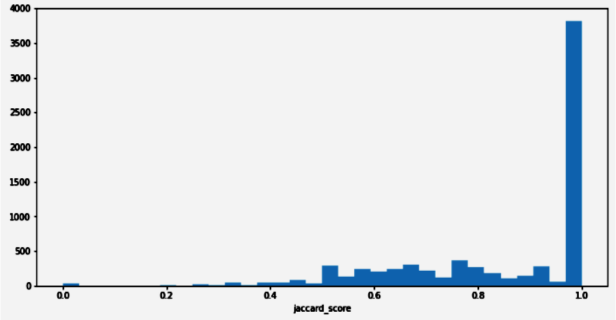
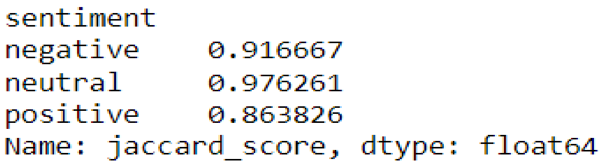


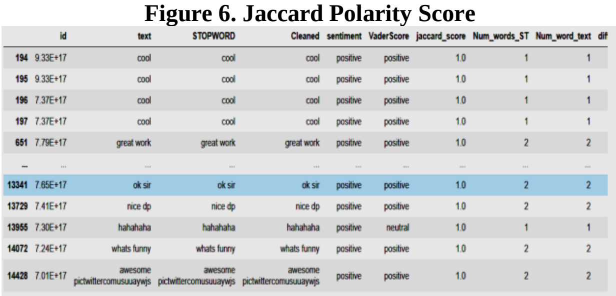
**Figure 5. Jaccard Score**

The Jaccard score plot shows that there is a peak for both the negative and positive plots around the score of 1. That is, the location of a cluster of tweets with high similarity between text and selected texts can represent accurate polarity. Hence, we can forecast text for selected texts for those tweets regardless of segment. One interesting approach is to look for tweets with fewer than three words in the text because the text might be used entirely as text. The following figure shows the polarity generated by the Jaccard score. We can observe that the text and the selected text are similar by having a more comprehensive look as depicted in the subsequent figure.



```
sentiment
negative    0.916667
neutral     0.976261
positive    0.863826
Name: jaccard_score, dtype: float64
```

**Figure 6. Jaccard Polarity Score**

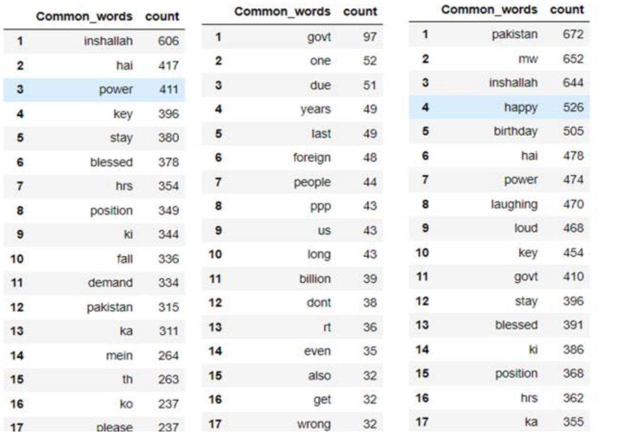


| | id | text | STOPWORD | Cleaned | sentiment | VaderScore | jaccard_score | Num_words_ST | Num_word_text | dif |
|---|---|---|---|---|---|---|---|---|---|---|
| 194 | 9.33E+17 | cool | cool | cool | positive | positive | 1.0 | 1 | 1 | |
| 195 | 9.33E+17 | cool | cool | cool | positive | positive | 1.0 | 1 | 1 | |
| 196 | 7.37E+17 | cool | cool | cool | positive | positive | 1.0 | 1 | 1 | |
| 197 | 7.37E+17 | cool | cool | cool | positive | positive | 1.0 | 1 | 1 | |
| 651 | 7.79E+17 | great work | great work | great work | positive | positive | 1.0 | 2 | 2 | |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| 13341 | 7.65E+17 | ok sir | ok sir | ok sir | positive | positive | 1.0 | 2 | 2 | |
| 13729 | 7.41E+17 | nice dp | nice dp | nice dp | positive | positive | 1.0 | 2 | 2 | |
| 13955 | 7.30E+17 | hahahaha | hahahaha | hahahaha | positive | neutral | 1.0 | 1 | 1 | |
| 14072 | 7.24E+17 | whats funny | whats funny | whats funny | positive | positive | 1.0 | 2 | 2 | |
| 14428 | 7.01E+17 | awesome pictwittercomusuuaywjs | awesome pictwittercomusuuaywjs | awesome pictwittercomusuuaywjs | positive | positive | 1.0 | 2 | 2 | |

**Figure 7. Detailed Jaccard Polarity and Sentiment Effect.**

As a result, we can observe that the most common words in Selected Text and Text are nearly identical, which was expected. Let's look at the most used terms in various sentiments. Words like getting, go, don't get, u, can't, lol, and like appear often in all three segments. That's odd since phrases like don't and can't have a negative connotation, yet words like lol have a good connotation. We can gain more insights on this after N-gram analysis A donut graph of unique positive, negative, and neutral words is depicted for comparison. It is interesting to see the word unique to different sentiments in the following order, we'll look at unique terms in each segment:

- Neutral
- Negative
- Positive



| | Common_words | count |
|---|---|---|
| 1 | inshallah | 606 |
| 2 | hai | 417 |
| 3 | power | 411 |
| 4 | key | 396 |
| 5 | stay | 380 |
| 6 | blessed | 378 |
| 7 | hrs | 354 |
| 8 | position | 349 |
| 9 | ki | 344 |
| 10 | fall | 336 |
| 11 | demand | 334 |
| 12 | pakistan | 315 |
| 13 | ka | 311 |
| 14 | mein | 264 |
| 15 | th | 263 |
| 16 | ko | 237 |
| 17 | please | 237 |

| | Common_words | count |
|---|---|---|
| 1 | govt | 97 |
| 2 | one | 52 |
| 3 | due | 51 |
| 4 | years | 49 |
| 5 | last | 49 |
| 6 | foreign | 48 |
| 7 | people | 44 |
| 8 | ppp | 43 |
| 9 | us | 43 |
| 10 | long | 43 |
| 11 | billion | 39 |
| 12 | dont | 38 |
| 13 | rt | 36 |
| 14 | even | 35 |
| 15 | also | 32 |
| 16 | get | 32 |
| 17 | wrong | 32 |

| | Common_words | count |
|---|---|---|
| 1 | pakistan | 672 |
| 2 | mw | 652 |
| 3 | inshallah | 644 |
| 4 | happy | 526 |
| 5 | birthday | 505 |
| 6 | hai | 478 |
| 7 | power | 474 |
| 8 | laughing | 470 |
| 9 | loud | 468 |
| 10 | key | 454 |
| 11 | govt | 410 |
| 12 | stay | 396 |
| 13 | blessed | 391 |
| 14 | ki | 386 |
| 15 | position | 368 |
| 16 | hrs | 362 |
| 17 | ka | 355 |

**Figure 8. Most Common Stop Words Neutral, Negative, and Positive Words (Left to right)**

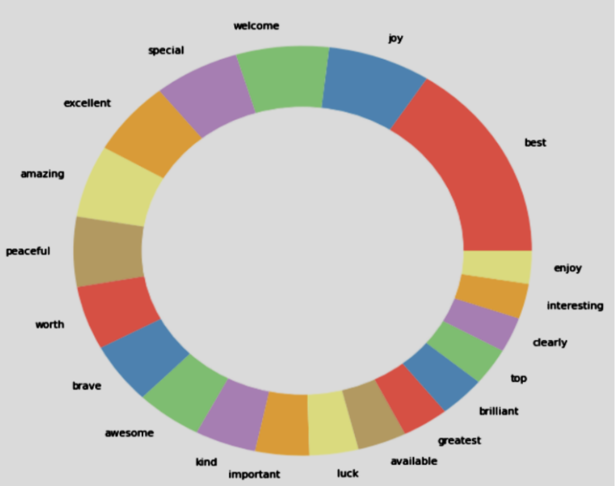


**Figure 9. Donut plot of Unique positive Words**

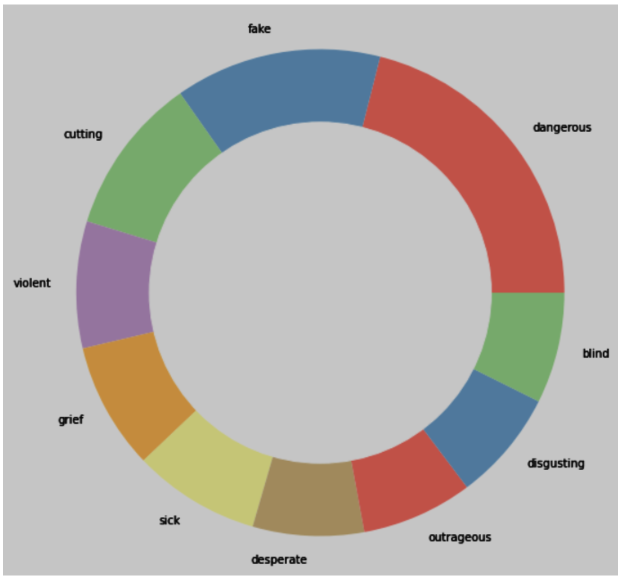


**Figure 10. Donut plot of Unique negative Words**

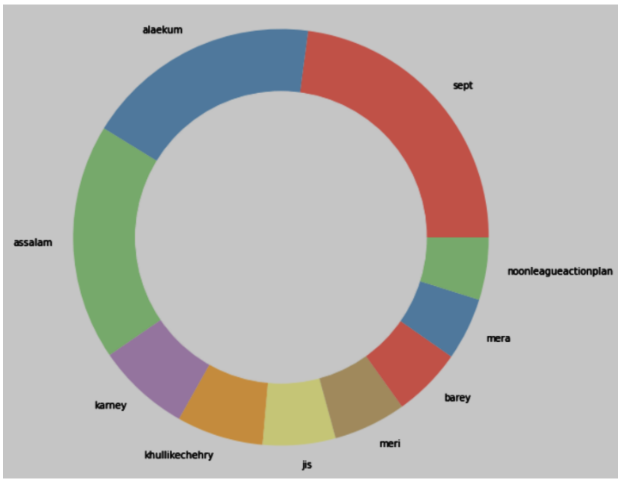


**Figure 11. Donut plot of Unique Neutral Words**

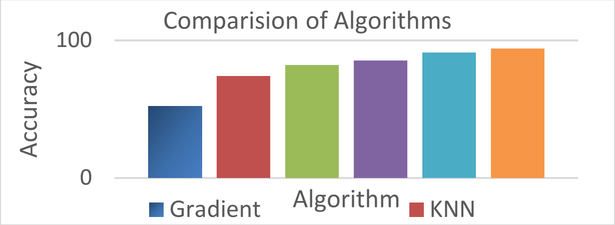
Comparison between different Classifiers on the same dataset by giving different training ranges of dataset is given in the table below. The table analyses and contrasts the total accuracy of seven supervised machine learning algorithms: K-Nearest Neighbour(K-NN) and Naive Bayes, Gradient Boosting, XGBoost, Logistic Regression, and Decision Tree. It was discovered that while Naive Bayes performed significantly better than other classifiers in the case of movie reviews, these methods performed as poorly in the case of hotel reviews.

**Table 1. Accuracy Comparison on different Test Datasets**

| **No. Of Experiments** | **No. Of Reviews** | **Accuracy** | | | | | |
|---|---|---|---|---|---|---|---|
| | | Naive Bayes | KNN | Gradient Boosting | XGBoost | Logistic Regression | Decision Tree |
| 1 | 100 | 56.78 | 43.11 | 43.35 | 51.04 | 63.22 | 76.15 |
| 2 | 200 | 64.29 | 41.26 | 4.097 | 58.64 | 52.94 | 46.37 |
| 3 | 500 | 70.06 | 42.56 | 41.42 | 64.11 | 48.25 | 77.05 |
| 4 | 1000 | 73.81 | 44.64 | 41.18 | 41.52 | 66.19 | 59.34 |
| 5 | 1500 | 77.23 | 48.21 | 42.01 | 42.05 | 55.74 | 51.11 |
| 6 | 2000 | 79.14 | 51.28 | 46.57 | 44.67 | 52.51 | 64.45 |
| 7 | 2500 | 79.82 | 52.03 | 47.04 | 55.48 | 55.04 | 79.61 |
| 8 | 3000 | 80.27 | 52.64 | 47.03 | 55.33 | 75.16 | 43.51 |
| 9 | 4000 | 82.11 | 53.92 | 49.75 | 64.11 | 46.55 | 78.91 |
| 10 | 4500 | 82.43 | 55.09 | 52.14 | 52.08 | 43.00 | 54.41 |

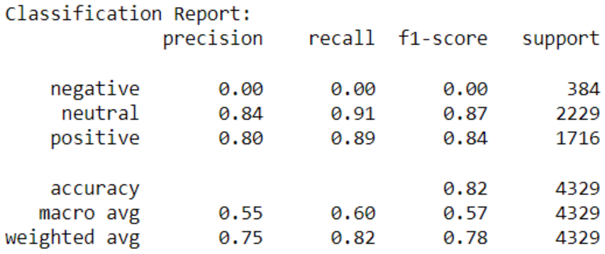
Likewise, we used a similar strategy to evaluate models on the Twitter data. The participating methods are Multinomial Naïve Bayes, KNN, Logistic Regression, Gradient Boosting, XGBoost, and Decision Tree. In our case, the Decision Tree Classifier outperforms the others. The comparison of accuracy is depicted in the following table.

| **Algorithms** | **Accuracy** |
|---|---|
| Gradient Boosting | 52% |
| KNN | 74% |
| Multinomial Naïve Bayes | 82% |
| Logistic Regression | 85% |
| XGBoost | 91% |
| Decision Tree | 94% |



### 4.1 Multinomial NB

The Trained Model's Classification Report for the given dataset:

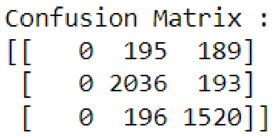


```
Classification Report:
              precision    recall  f1-score   support

    negative       0.00      0.00      0.00       384
     neutral       0.84      0.91      0.87      2229
    positive       0.80      0.89      0.84      1716

    accuracy                           0.82      4329
   macro avg       0.55      0.60      0.57      4329
weighted avg       0.75      0.82      0.78      4329
```

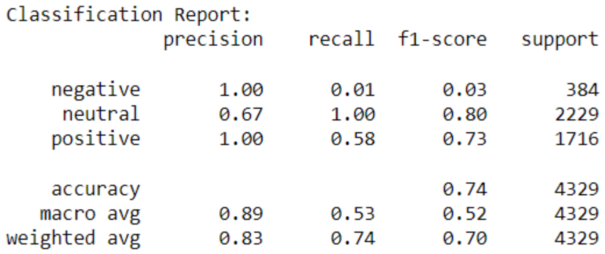
The trained model's Confusion Matrix Report for the provided dataset:



```
Confusion Matrix :
[[   0  195  189]
 [   0 2036  193]
 [   0  196 1520]]
```

### 4.2 KNN

The Trained Model's Classification Report for the given dataset:

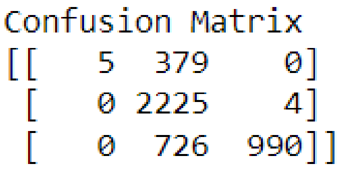


```
Classification Report:
              precision    recall  f1-score   support

    negative       1.00      0.01      0.03       384
     neutral       0.67      1.00      0.80      2229
    positive       1.00      0.58      0.73      1716

    accuracy                           0.74      4329
   macro avg       0.89      0.53      0.52      4329
weighted avg       0.83      0.74      0.70      4329
```

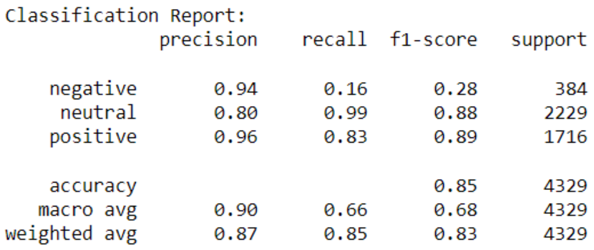
The trained model's Confusion Matrix Report for the provided dataset:



```
Confusion Matrix
[[   5  379    0]
 [   0 2225    4]
 [   0  726  990]]
```

### 4.3 Logistic Regression

This is the Trained Model's Classification Report for the given dataset:

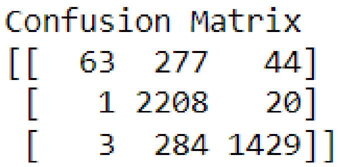


```
Classification Report:
              precision    recall  f1-score   support

    negative       0.94      0.16      0.28       384
     neutral       0.80      0.99      0.88      2229
    positive       0.96      0.83      0.89      1716

    accuracy                           0.85      4329
   macro avg       0.90      0.66      0.68      4329
weighted avg       0.87      0.85      0.83      4329
```

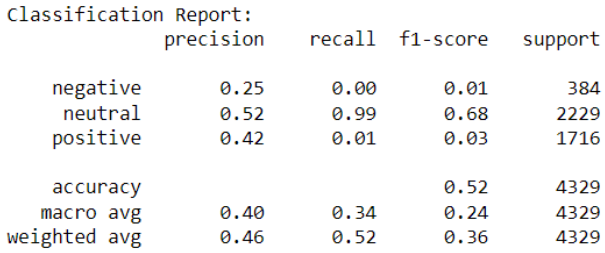
The trained model's Confusion Matrix Report for the provided dataset:



```
Confusion Matrix
[[  63  277   44]
 [   1 2208   20]
 [   3  284 1429]]
```

### 4.4 Gradient Boosting

The Trained Model's Classification Report for the given dataset:

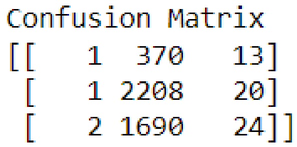


```
Classification Report:
              precision    recall  f1-score   support

    negative       0.25      0.00      0.01       384
     neutral       0.52      0.99      0.68      2229
    positive       0.42      0.01      0.03      1716

    accuracy                           0.52      4329
   macro avg       0.40      0.34      0.24      4329
weighted avg       0.46      0.52      0.36      4329
```

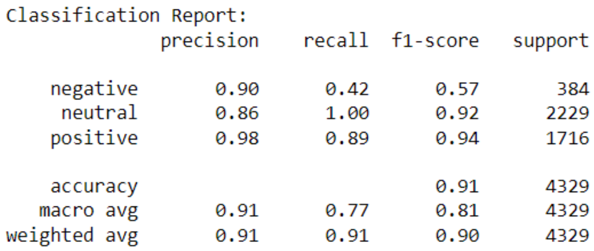
The trained model's Confusion Matrix Report for the provided dataset:



```
Confusion Matrix
[[   1  370   13]
 [   1 2208   20]
 [   2 1690   24]]
```

### 4.5 XGB

The Trained Model's Classification Report for the given dataset:

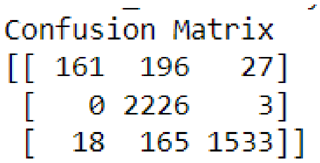


```
Classification Report:
              precision    recall  f1-score   support

    negative       0.90      0.42      0.57       384
     neutral       0.86      1.00      0.92      2229
    positive       0.98      0.89      0.94      1716

    accuracy                           0.91      4329
   macro avg       0.91      0.77      0.81      4329
weighted avg       0.91      0.91      0.90      4329
```

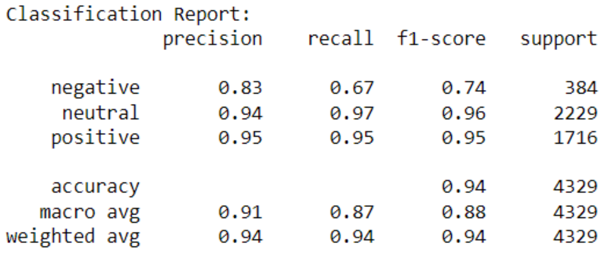
The trained model's Confusion Matrix Report for the provided dataset:



```
Confusion Matrix
[[ 161  196   27]
 [   0 2226    3]
 [  18  165 1533]]
```

### 4.7 Decision Tree

The Trained Model's Classification Report for the given dataset:

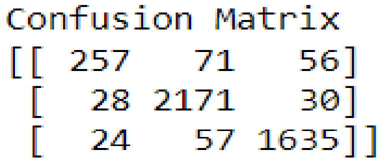


```
Classification Report:
              precision    recall  f1-score   support

    negative       0.83      0.67      0.74       384
     neutral       0.94      0.97      0.96      2229
    positive       0.95      0.95      0.95      1716

    accuracy                           0.94      4329
   macro avg       0.91      0.87      0.88      4329
weighted avg       0.94      0.94      0.94      4329
```

This is the trained model's Confusion Matrix Report for the provided dataset:



```
Confusion Matrix
[[ 257   71   56]
 [  28 2171   30]
 [  24   57 1635]]
```

## V. CONCLUSION & FUTURE WORK

This work explores various pre-processing approaches that affect the classification of the Twitter polarity of feelings. Experimental results suggest a minimum impact on the performance of the classifier by removing URLs, deleting stop words, and deleting numerals. For the Twitter sentiment classification job, we identify acceptable pre-processing approaches and functional models for several classifiers. Thus, the basic knowledge aimed at analyzing Twitter sentiments is clearly described in this study. The accuracy/result of each method allows us to imagine the efficiency of the given technique. In this study, we have shown a system for extracting knowledge from tweets and then classifying tweets based on their semantics. The research aims to classify a large Twitter data corpus into sets of feelings, positive and negative, and neutral. Higher accuracy is attained by incorporating aspects of feeling rather than conventional text classification. We will examine numerous related challenges in the future. Effective uncontrolled or lexical classifiers, domain adaptation, and selection of features are all key problems requiring more investigation. On the other hand, we want to examine how it can be implemented to solve real-world problems with an efficient sentiment analysis algorithm. Predicting presidential elections, evaluating reputations for products and film ratings, etc. We would also explore the engineering elements of Twitter sentiment analysis.

## VI. REFERENCES

[1] Jones, C., The LGBTQ Social Media Problem: Facebook Case Studies. 2019.

[2] Iftikhar, R. and M.S. Khan, social media big data analytics for demand forecasting: development and case implementation of an innovative framework, in Research Anthology on Big Data Analytics, Architectures, and Applications. 2022, IGI Global. p. 902-920.

[3] Kathuria, A., et al., Classifying the user intent of web queries using k-means clustering. Internet Research, 2010.

[4] Appel, G., et al., The future of social media in marketing. Journal of the Academy of Marketing Science, 2020. 48(1): p. 79-95.

[5] Korenek, P. and M. Šimko, Sentiment analysis on microblog utilizing appraisal theory. World Wide Web, 2014. 17(4): p. 847-867.

[6] Batool, R., et al. Precise tweet classification and sentiment analysis. in 2013 IEEE/ACIS 12th International Conference on Computer and Information Science (ICIS). 2013. IEEE.

[7] Ellison, N.B., C. Steinfield, and C. Lampe, The benefits of Facebook "friends:" Social capital and college students use of online social network sites. Journal of computer-mediated communication, 2007. 12(4): p. 1143-1168.

[8] Morgan, N.A., C.S. Katsikeas, and D.W. Vorhies, Export marketing strategy implementation, export marketing capabilities, and export venture performance. Journal of the academy of marketing science, 2012. 40(2): p. 271-289.

[9] Xia, R., C. Zong, and S. Li, Ensemble of feature sets and classification algorithms for sentiment classification. Information sciences, 2011. 181(6): p. 1138-1152.

[10] Seman, N. and N.A. Razmi is, Machine learning-based technique for big data sentiments extraction. IAES International Journal of Artificial Intelligence, 2020. 9(3): p. 473.

[11] Ahmad, M., S. Aftab, and I. Ali, Sentiment analysis of tweets using svm. Int. J. Comput. Appl, 2017. 177(5): p. 25-29.

[12] Watanabe, H., M. Bouazizi, and T. Ohtsuki, Hate speech on Twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection. IEEE access, 2018. 6: p. 13825-13835.

[13] Singh, D., et al., Changing trends in the use of kratom (Mitragyna speciosa) in Southeast Asia. Human Psychopharmacology: Clinical and Experimental, 2017. 32(3): p. e2582

[14] Barbosa, L. and J. Feng. Robust sentiment detection on twitter from biased and noisy data. in Coling 2010: Posters. 2010.

[15] Gamon, M. Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. in COLING 2004: Proceedings of the 20th international conference on computational linguistics. 2004.

[16] Wang, X., et al. Non-local neural networks. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[17] Zimbra, D., M. Ghiassi, and S. Lee. Brand-related Twitter sentiment analysis using feature engineering and the dynamic architecture for artificial neural networks. in 2016 49th Hawaii International Conference on System Sciences (HICSS). 2016. IEEE.

[18] Godbole, N., M. Srinivasaiah, and S. Skiena, Large-Scale Sentiment Analysis for News and Blogs. Icwsm, 2007. 7(21): p. 219-222.

[19] Lösch, U., S. Bloehdorn, and A. Rettinger. Graph kernels for RDF data. in Extended semantic web conference. 2012. Springer.

[20] Tumasjan, A., et al. Predicting elections with twitter: What 140 characters reveal about political sentiment. in Proceedings of the International AAAI Conference on Web and Social Media. 2010.

[21] Zhao, N., et al., Evaluating the validity of simplified Chinese version of LIWC in detecting psychological expressions in short texts on social network services. PLoS One, 2016. 11(6): p. e0157947.

[22] Jiang, H., X. Zhang, and J.D. Choi. Automatic text-based personality recognition on monologues and multiparty dialogues using attentive networks and contextual embeddings (student abstract). in Proceedings of the AAAI Conference on Artificial Intelligence. 2020.

[23] Gottipati, S., V. Shankararaman, and S. Gan. A conceptual framework for analyzing students' feedback. in 2017 IEEE Frontiers in Education Conference (FIE). 2017. IEEE.

[24] Soldatova, L.N., et al., EXACT2: the semantics of biomedical protocols. BMC bioinformatics, 2014. 15(14): p. 1-11.