Sentiment Analysis Using Machine Learning Classifiers: Evaluation of Performance

R Vidhya[1], Pavithra Gopalakrishnan[2], Nanda Kishore Vallamkondu[3]
{vidhyar@srmist.edu.in[1], pavi.gk99@gmail.com[2], vallamnanda12@gmail.com[3]}

School of Computing, SRM Institute of Science and Technology, Kattankulathur, Chennai, 603203 Tamil Nadu, India.[1,2,3]

Abstract. Sentiment analysis is a process that is a very popular concept nowadays because of the high volume of reviews, micro blogs, comments etc., generated in different sites like e-commerce and social networking sites. The main problem in the current system is, for users to know the polarity result of bulk data, which is very tough because users need to study and understand each review in terms of the polarity. Users are expecting a onetime result of the polarity of bulk reviews, comments, micro blogs etc. In social networking sites, users post their status or opinions to share to the world. In this category, Twitter is the most popular one. In twitter, users post many micro blogs related to a topic or crisis etc, and the topic may be linked with a greater number of micro blogs based on the keywords or hash tags used. In twitter, we can search for any topic with keywords or hash tags, and we get a bulk of responses of the users world-wide. If we want to know the exact opinions of the users, we need to analyze the data with sentiment analysis. Sentiment analysis is a concept of defining a statement as positive, neutral, or negative by analyzing words of the statement. Many concepts have been proposed for this requirement, and many sentidatasets have been prepared for this requirement. But by taking the advantages of Machine Learning we are proposing a concept of sentiment analysis in twitter using ML techniques. In this we use multiple ML techniques such as Random Forest, Naïve Bayes and Support Vector Machine for evaluation and comparison of the results.

Keywords: SVM, NB, TF-IDF, Sentidatasets

1 Introduction

In the past decade, there has been no limit to the range of information that is being conveyed using tweets and text messages, which are often short messages used for sharing opinions called sentiments about things happening around them. The language used is mostly informal with creative spellings, new words, URLs, abbreviations, punctuations, and hashtags, which is a way of tagging. Similarly, it uses Pre-Processing, text analysis and computational linguistics for identifying and analyzing the raw data and predicting opinions. It is also referred as opinion mining. The internet has established a platform for people to express their views, emotions on products, people, and things around them. The objective of sentiment analysis is to extract important insights from large amounts of data by removing unnecessary data that would help organizations, better decision making and quality product consumption. In this paper, we look at an approach, where a model is built for classifying tweets, which are

I3CAC 2021, June 07-08, Chennai, India
Copyright © 2021 EAI
DOI 10.4108/eai.7-6-2021.2308565

retrieved using consumer keys, access tokens from TwitterSearch API. This process is for calculating polarity.

2 Related Work

Social Media (SM) are online applications which enable sharing the moods, status, opinions of users to their virtual social circles for example twitter, Facebook, etc. [1], [2]. In machine learning there are few topics which are used to build the requirements of prediction, aggregation, and clustering etc. In different surveys many systems were proposed for sentiment analysis by using basic concepts of N-grams, string comparisons, clustering etc. Those only provide limited solutions of sentiment analysis. Mainly, sentiment analysis was used to produce important results like political analysis, rating prediction, product reviews monitoring etc. [3] [4]. Most surveys depend on a static sentiwordnet dataset to find the sentiment analysis. But we require finding a proper solution to find the polarity of the tweets.

2.1 Scope of the Work

The scope of work includes the following:

- The understanding of public opinions' importance in terms of a particular topic, and that opinion's collection from the internet using social networking sites.
- We can understand the importance of the Machine Learning classification algorithms and how it helps us in text classification.

Based on the training data of text labeled data, we will understand how classification algorithms can process using tokenizer concepts and classify the results.

3 Implementation

3.1 Architecture

In our system, sentiment analysis is based on machine learning algorithms applied to the tweets, and the architecture is developed in two modules. One is the Admin module, and the other one is a User module. In the following architecture diagram, we represent admin flow and user flow in two formats. Admin flow is represented in brown color and user flow is represented in blue color. In the admin module we are using three machine learning classifiers for training the data, and in the testing module we compare the algorithms with accuracy scores. We chose the best algorithm in terms of accuracy and deployed in the user module for performing the sentiment analysis of real time tweets. In this chapter we will discuss our main and sub modules in detail.

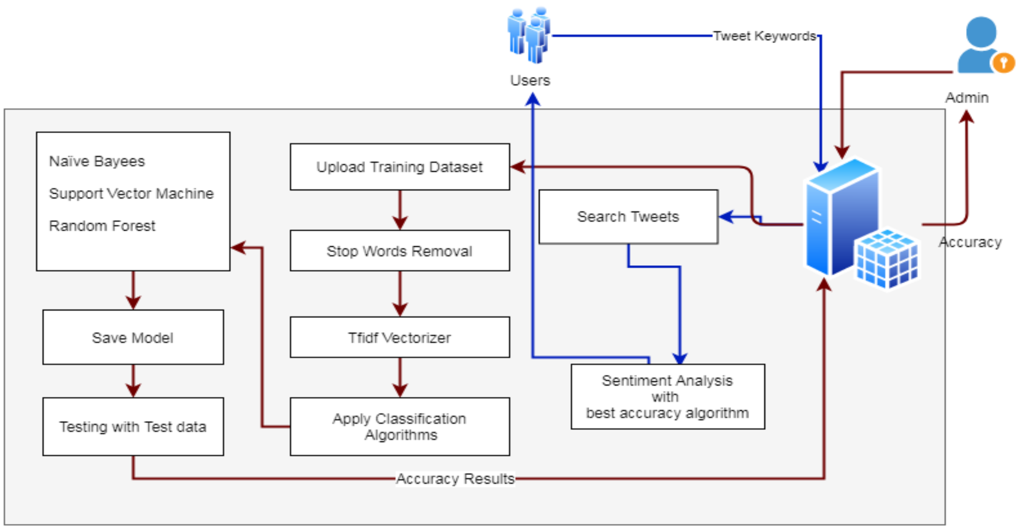


Fig. 1 System Architecture

3.2 Admin Module

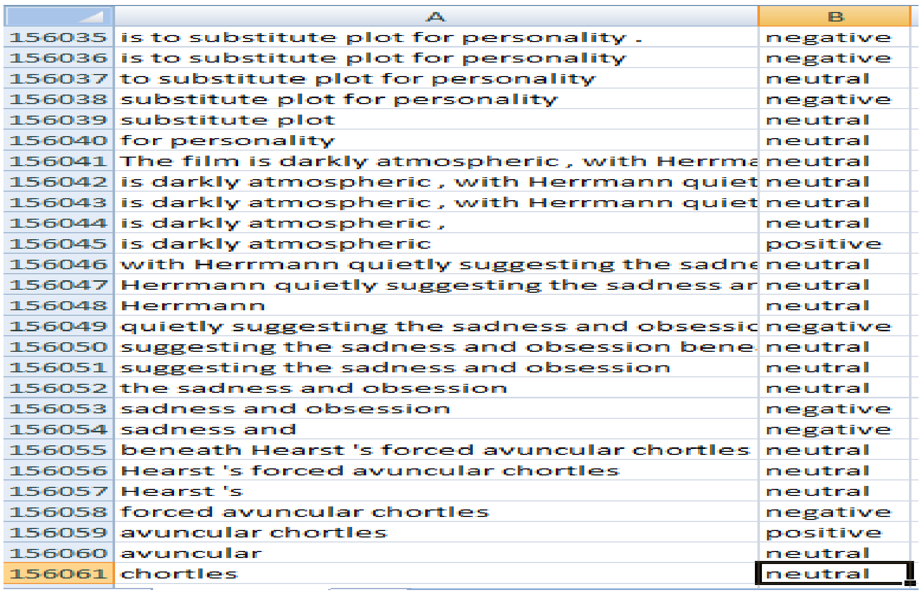
In this admin module we train and test the dataset using three machine learning classifiers for accuracy calculation comparison.

3.3 Upload Dataset

For theSentiment analysis we have taken the dataset which consists of 156062 rows of labeled data. In this, we have words and statements in one column and in the other column we have label class of sentiment polarity, namely positive, negative and neutral. In the following table we can see the dataset description.

Table 1. Description of the dataset

String Data	Classes: negative, neutral and positive



	A	B
156035	is to substitute plot for personality .	negative
156036	is to substitute plot for personality	negative
156037	to substitute plot for personality	neutral
156038	substitute plot for personality	negative
156039	substitute plot	neutral
156040	for personality	neutral
156041	The film is darkly atmospheric , with Herrma	neutral
156042	is darkly atmospheric , with Herrmann quiet	neutral
156043	is darkly atmospheric , with Herrmann quiet	neutral
156044	is darkly atmospheric ,	neutral
156045	is darkly atmospheric	positive
156046	with Herrmann quietly suggesting the sadne	neutral
156047	Herrmann quietly suggesting the sadness ar	neutral
156048	Herrmann	neutral
156049	quietly suggesting the sadness and obsessic	negative
156050	suggesting the sadness and obsession bene	neutral
156051	suggesting the sadness and obsession	neutral
156052	the sadness and obsession	neutral
156053	sadness and obsession	negative
156054	sadness and	negative
156055	beneath Hearst 's forced avuncular chortles	neutral
156056	Hearst 's forced avuncular chortles	neutral
156057	Hearst 's	neutral
156058	forced avuncular chortles	negative
156059	avuncular chortles	positive
156060	avuncular	neutral
156061	chortles	neutral

Fig 2. Screenshot of the dataset

3.4 Stop words Removal

This step comes under the process of feature extraction. In every review statement of the training dataset, we identified the stop words and removed it. In the API which we are using we have a list which consists of list of English stop words like 'a', 'about', 'according', 'across' etc. We remove stop words because these words are useless for any judgment of text classification, and we can achieve better database space and computation cost.

3.5 TfidfVectorizer

Generally, Machine Learning algorithm only deals with numeric format data, hence in text classification we need to convert the data into numeric vector format. Generally, this process can be done in two ways, one is CountVectorizer and the other one is TfidfVectorizer. In CountVectorizer it will take the word count of the text, it is a very basic process. But in the TfidfVectorizer it will take the Tf*IDf score as its numerical data for the vector model.

For example,

dataset=*['problem of devil', 'devil queen', 'horizon problem']*

Table 2. TfidfVectorizer Example

doc	*Devil*	*horizon*	*Of*	*problem*	*queen*
0	*0.517*	*0.000*	*0.680*	*0.517*	*0.000*
1	*0.605*	*0.000*	*0.000*	*0.000*	*0.795*
2	*0.000*	*0.795*	*0.000*	*0.605*	*0.000*

3.6 Apply Classification Algorithms

In this sub module after conversion of the dataset into the TfidfVectorizer, we apply the following classification algorithms:

- Naive Bayes
- Random Forest
- Support Vector Machine

Naive Bayes Algorithm

We upload the dataset (.csv file) by using the pandas API in python and we apply the TfidfVectorizer. After that we train the dataset and save the model file in the '.sav' file format. We store the fit model in .sav format for future testing purposes.

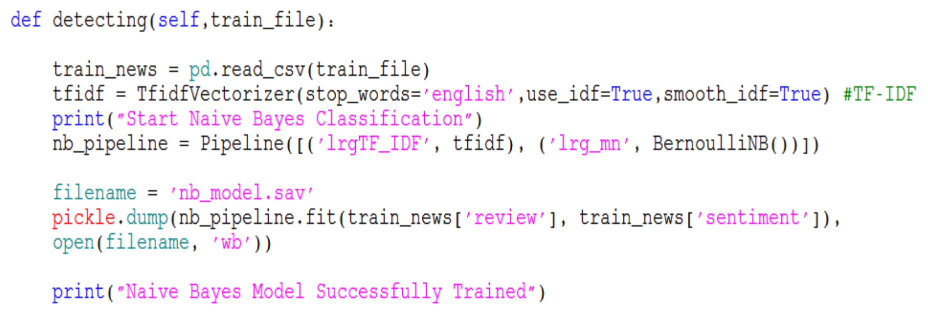
Naïve algorithm is an adoption format of the Bayes Formula. By using this we can calculate the probability of each class in terms of the attributes. We derive naïve bayes formula as the following:

$$p(c/d) = \frac{P\left(\frac{d}{c}\right)P(c)}{P(d)} \quad (1)$$

Where in:

'c' is a class,
'd' is a document,
'P(c)' is a class probability,

In the following figure, we can see the implementation code of training the dataset using Naive bayes algorithm.



```
def detecting(self,train_file):

    train_news = pd.read_csv(train_file)
    tfidf = TfidfVectorizer(stop_words='english',use_idf=True,smooth_idf=True) #TF-IDF
    print("Start Naive Bayes Classification")
    nb_pipeline = Pipeline([('lrgTF_IDF', tfidf), ('lrg_mn', BernoulliNB())])

    filename = 'nb_model.sav'
    pickle.dump(nb_pipeline.fit(train_news['review'], train_news['sentiment']),
    open(filename, 'wb'))

    print("Naive Bayes Model Successfully Trained")
```

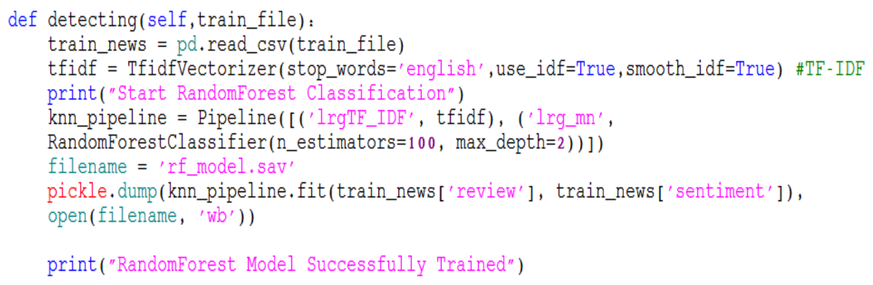
Fig 3. Sample code of Naive Bayes

Random Forest Algorithm

We upload the dataset (.csv file) by using the pandas API in python and we apply the TfidfVectorizer. After that we train the dataset and save the model file in the '.sav' file format. We store the fit model in .sav format for future testing purposes.

Random Forest algorithm is a supervised algorithm for both classification and regression algorithm. It computes the decision based on the highest scores of multiple decision trees.

In the following figure, we can see the implementation code of the training the dataset using Random Forest algorithm.



```
def detecting(self,train_file):
    train_news = pd.read_csv(train_file)
    tfidf = TfidfVectorizer(stop_words='english',use_idf=True,smooth_idf=True) #TF-IDF
    print("Start RandomForest Classification")
    knn_pipeline = Pipeline([('lrgTF_IDF', tfidf), ('lrg_mn',
    RandomForestClassifier(n_estimators=100, max_depth=2))])
    filename = 'rf_model.sav'
    pickle.dump(knn_pipeline.fit(train_news['review'], train_news['sentiment']),
    open(filename, 'wb'))

    print("RandomForest Model Successfully Trained")
```

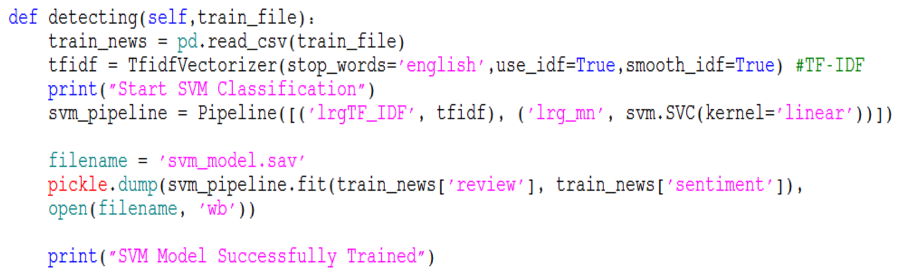
Fig 4. Sample code of Random Forest

Support Vector Machine

We upload the dataset (.csv file) by using the pandas API in python and we apply the TfidfVectorizer. After that we train the dataset and save the model file in the '.sav' file format. We store the fit model in .sav format for future testing purposes.

SVM algorithm is a supervised algorithm for both classification and regression algorithm. It calculate distance between the difference of the two classes by defining the closest points.

In the following figure, we can see the implementation code of the training the dataset using SVM algorithm.



```
def detecting(self,train_file):
    train_news = pd.read_csv(train_file)
    tfidf = TfidfVectorizer(stop_words='english',use_idf=True,smooth_idf=True) #TF-IDF
    print("Start SVM Classification")
    svm_pipeline = Pipeline([('lrgTF_IDF', tfidf), ('lrg_mn', svm.SVC(kernel='linear'))])

    filename = 'svm_model.sav'
    pickle.dump(svm_pipeline.fit(train_news['review'], train_news['sentiment']),
    open(filename, 'wb'))

    print("SVM Model Successfully Trained")
```

Fig 5. Sample code of SVM

3.7 Save Model

After executing our three classification algorithms, we can see the nb_model.sav, rf_model.sav and svm_model.sav files in the current folder. These files can be used for testing the data n number of times without training the dataset again. Simply, we store the fit model in to a file. And for the 156062 rows of the data, we convert it to .sav file with different sizes. For Naïve bayes algorithm, 2mb size of save file is created, for Random Forest 1mb size of save file is created and for SVM 57mb size of save file is created.

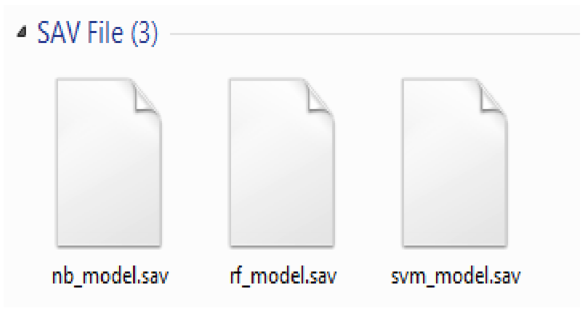


Fig 6. Screenshot of sav files

3.8 Testing with test data

After creation of the model files of three algorithms we take testing file (.csv) that consists of 1000 records. This data is already labeled data. After getting the results from the individual algorithms we compare with actual results. Then we can calculate the accuracy score by using the following formula:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \quad (2)$$

where,

TP: True Positive
TN: True Negative
FP: False Positive
FN: False Negative

In this testing module we have Support Vector Machine algorithm with the best score in accuracy.

3.9 User Module

In this user module, user can provide keywords or hashtags of a topic, then user will get the real time tweets or re-tweets published in twitter which are identified by searching for the keyword in tweets and then display the evaluated polarity of the tweets as positive or negative or neutral.

3.10 Search Tweets

In this module, user searches for tweets using keywords or hash tags from the real time twitter database by using python API. In this module we will describe the process of getting the tweets from twitter using API's. We are developing our proposed system application by using Python coding language. So, we are using the python API called Tweepy for getting the tweets related to a topic.

3.11 Sentiment Analysis

In this module, we choose the best algorithm in terms of the accuracy and deploy it in the user module for performing sentiment analysis of the real time tweets. In this testing module, Support Vector Machine algorithm shows best score in accuracy. So, our system will give the one-shot result of the polarity results of the tweets.

4 Experimental Results

4.1 Accuracy Score Graph

In the testing result page, we have a user interface to view the accuracy graph.

Table 3. Accuracy Results

Random Forest	Naïve Bayes	Support Vector Machine
0.67	0.731	0.783

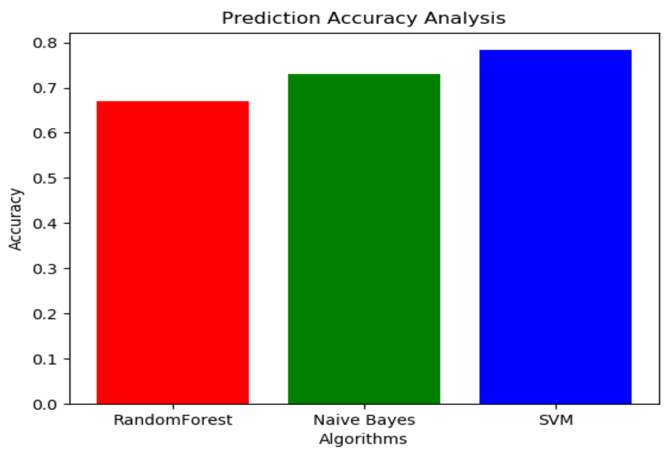


Fig 7. Accuracy Graph

4.2 F1Score Graph

In the testing result page, we have a user interface to view the F1socre graph.

Table 4. F1 Score Results

Random Forest	Naïve Bayes	Support Vector Machine
0.54	0.71	0.77

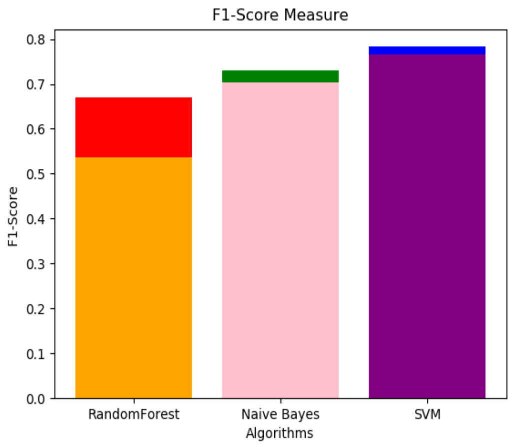
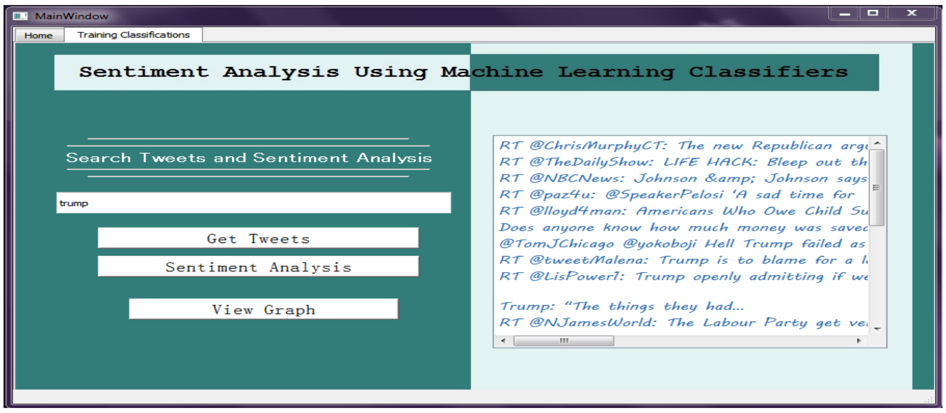


Fig 8.F1Score Graph

4.3 Search tweets in user home page

After successful login of the user, user can see 'get tweets option', user can enter the keywords and get the tweets. In the following example 'trump' has been inputted as keywords. We can see the tweets in right side.


MainWindow
Home
Training Classifications
Sentiment Analysis Using Machine Learning Classifiers
Search Tweets and Sentiment Analysis
trump
Get Tweets
Sentiment Analysis
View Graph

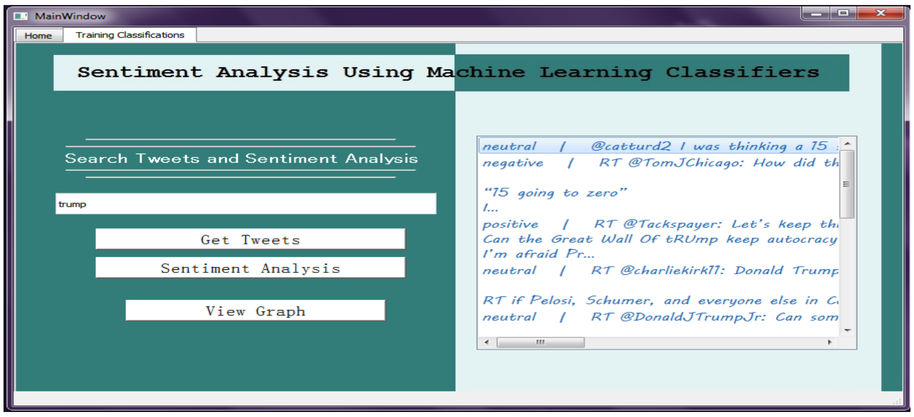
```
RT @ChrisMurphyCT: The new Republican arg
RT @TheDailyShow: LIFE HACK: Bleep out th
RT @NBCNews: Johnson &amp; Johnson says
RT @paz4u: @SpeakerPelosi 'A sad time for
RT @lloyd4man: Americans Who Owe Child Su
Does anyone know how much money was save
@TomJChicago @yokoboji Hell Trump failed as
RT @tweetMalena: Trump is to blame for a l
RT @LisPower1: Trump openly admitting if we

Trump: "The things they had...
RT @NJamesWorld: The Labour Party get ve
```

Fig 9. Search Tweets

4.4 Sentiment Analysis process

In the result page we have shown the process of the sentiment analysis. By clicking on the Sentiment Analysis button, SVM model will be implemented and all tweets will classified and labeled with sentiment classes like positive, negative, or neutral.


MainWindow
Home
Training Classifications
Sentiment Analysis Using Machine Learning Classifiers
Search Tweets and Sentiment Analysis
trump
Get Tweets
Sentiment Analysis
View Graph

```
neutral   |   @catturd2 I was thinking a 15
negative   |   RT @TomJChicago: How did th

“15 going to zero”
I...
positive   |   RT @Tackspayer: Let’s keep th
Can the Great Wall Of tRUmp keep autocracy
I’m afraid Pr...
neutral   |   RT @charliekirk11: Donald Trump

RT if Pelosi, Schumer, and everyone else in C
neutral   |   RT @DonaldJTrumpJr: Can som
```

Fig 10. Sentiment Analysis Process

Sentiment Results graph

We can view the results graph by clicking on the ‘View Graph’ option. We can compare the sentiment score values of tweets as how many neutral, how many positive, and how many negative tweets are there, graphically.

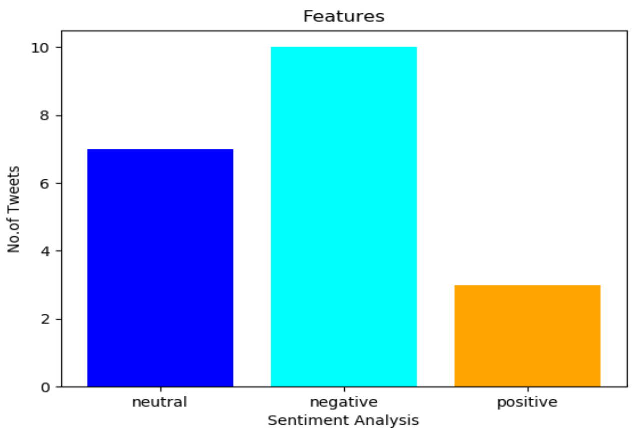


Fig 11. Sentiment results Graph

5 Conclusions

In our system we performed sentiment analysis by using machine learning algorithms on the tweets. Sentiment analysis is a process of calculating the polarity of the statement as to get the polarity namely positive, negative and neutral, of the posts, micro blogs, opinions or feedback which are published in social media sites or E-commerce sites. We developed the architecture in two modules. One is Admin module, and another one is User module. In admin module we are using three machine learning classifiers for training the data, in testing module we can compare the algorithms with accuracy scores. In our testing we see that Support

Vector Machine algorithm shows best accuracy when compared with Naive Bayes and Random Forest algorithms. We chose SVM algorithm based on the accuracy and deployed it in the user module for performing sentiment analysis of the real time tweets

References

[1] S. Rosenthal, P. Nakov, S. Kiritchenko, S. Mohammad, A. Ritter, and V. Stoyanov, "Semeval-2015 task 10: Sentiment analysis in twitter," in Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015), pp. 451–463, 2015.

[2] Y. Wan and Q. Gao, "An ensemble sentiment classification system of twitter data for airline services analysis," in Data Mining Workshop (ICDMW), 2015 IEEE International Conferenceon, pp. 1318–1325, IEEE, 2015.

[3] Y. Wan and Q. Gao, "An ensemble sentiment classification system of twitter data for airline services analysis," in Data Mining Workshop (ICDMW), 2015 IEEE International Conferenceon, pp. 1318–1325, IEEE, 2015.

[4] Z. Jianqiang and G.Xiaolin, "Comparison research on text pre- processing methods on twitter sentiment analysis," IEEE Access, vol. 5, pp. 2870–2879, 2017.