EPIDEMIC OUTBREAK PREDICTION USING ARTIFICIAL INTELLIGENCE

Nimai Chand Das Adhikari, Arpana Alka, Vamshi Kumar Kurva, Suhas S, Hitesh Nayak, Kumar Rishav, Ashish Kumar Nayak, Sankalp Kumar Nayak, Vaisakh Shaj, Karthikeyan, Srikant Nayak

Analytic Labs Research Group

ABSTRACT

Intelligent Models for predicting diseases whether building a model to help the doctor or even preventing its spread in an area globally, is increasing day by day. Here we present a noble approach to predict the disease prone area using the power of Text Analysis and Machine Learning. Epidemic Search model using the power of the social network data analysis and then using this data to provide a probability score of the spread and to analyse the areas whether going to suffer from any epidemic spread-out, is the main focus of this work. We have tried to analyse and showcase how the model with different kinds of pre-processing and algorithms predict the output. We have used the combination of words-n grams, word embeddings and TF-IDF with different data mining and deep learning algorithms like SVM, Naïve Bayes and RNN-LSTM. Naïve Bayes with TF-IDF performed better in comparison to others.

KEYWORDS

Natural Language Processing, Text Mining, Text Analysis, Support Vector Machines, LSTM, Naive Bayes, Text Blob, Tweet Sentiment Analysis

1. INTRODUCTION

The power of predictive modelling is gaining the importance in this ever-growing research areas. Researchers are taking the advantage of the power of analytics to analyse the data and create a relation out of it. Text Analysis with the twitter data is one of the most important area of focus for all the researchers. One of the text analysis problem is the sentiment analysis. It has been a useful tool for analysing array of problems that are related to human computer interaction. This can be extended to the fields of sociology, advertising, healthcare [15] and marketing, and thus ultimately to the area of social media. Sentiment could be described as subjective nature of an individual, which relates to the "private state" of an individual. Private state could be described as something which cannot be classified either as an objective observation or for verification. Sentiment analysis is an important field of research that has been closely associated with natural language processing, computational linguistics and text mining, which is used during sentiment analysis to identify the information and extract the same. The extracted information is quantified, then affective states are studied along with the subjective information. It could be said that sentiment analysis aims to ascertain the attitude of the individual, from whom the information is generated, with respect to the contextual polarity of the content which is being analysed [1]. Sentiment analysis is also referred to as "subjective analysis" or "opinion mining", along with traces of connectivity to affective computing. Affective computing refers to the recognition of emotions by computers (Picard, 2000).

1.1 Sentiment Analysis

Sentiment analysis usually studies those elements which are subjective in nature. These elements are the words, phrases, or on some occasions it might be the sentences. This shows that sentiments exist in the form of small linguistic units. Through sentiment analysis, one can ascertain the actual intent of the author. Hence, this could be described as a phenomenon which is

capable of detecting the sentiments from a given content automatically. Indeed, the sentiment analysis has been a boon for the organizations, because through opinion mining organizations are capable of making better decisions than before. In this case, an organization can develop its strategy based on the analysis that is derived from the opinion of the users. Hence, this facilitates better decision making by deciphering the emotions that is embedded in the word or a sentence (Pozzi, et al., 2016). It is a known fact, that sentiment analysis can be used effectively for extracting sentiments from contents displayed on social media. However, there are multiple research works which has demonstrated that this phenomenon can be used effectively to counter the epidemics. One of the research done in this regard, demonstrated that there is a strong relationship between the frequency of social media messages and the online news articles. The epidemic in question for this research was "measles". The research demonstrated, how monitoring of social media can be effectively used for the improvement of communication policies that can create general awareness amongst the masses. The data that has been extracted from the content of social media provide deeper insights into the "opinion" of the public which are at a certain moment, are salient amongst the public, that actually assists the health institutes to respond on an immediate basis to the concerns of public. In other words, through sentiment analysis opinion of the public related to epidemic disease can be sensed, and appropriate action can be taken based on that [4]. Opinion mining of social media content through sentiment analysis helps the public health officials to keep a track of spreading epidemics and take counter measure accordingly. They can also track the locations where the epidemic is spreading. Moreover, through sentiment analysis of the contents of the social media, it will be easier to detect the speed at which the epidemic is spreading. In another research it was found that social media platforms like twitter can be used as an important source of information, in a real-time situation. This helps to understand, how much concerned public is, on the outbreak of epidemic. This can be thoroughly achieved by sentiment classification of the twitter messages to develop an understanding on "degree of concern (DOC)", that is exhibited by the twitter users. The research adopts two-step process for classifying the sentiments, identifying the personal tweets and the negative tweets separately. With the help of this workflow, the researcher developed a tool for monitoring epidemic sentiments, that will visualize the concerns of the users of the twitter, regarding different types of epidemics. In this regard clue-based learning methods and machine learning method were used for classification of the twitter messages. With the help of Multinomial Naïve Bayes method, a classifier was built, and was sentiment analysis of tweets (Ji, et al., 2013). This phenomenon has been also classified as "knowledge-based tweet classification for the sentiment monitoring of diseases. For sentiment analysis of epidemics, the investigation of the sentiment dynamics of the media sources needs to be done primarily. Here, the media sources include tweeter and different online news publications which publishes content on the outbreak of epidemic diseases. A generic approach to perform the sentiment analysis will be as discussed in (Kim, et al., 2015).

1.2 Approaches

There are multiple approaches that has been devised to detect the outbreak of epidemics through twitter. One of the most common approach that is used to create a locational network for a specific country is completely based on the data taken from twitter. The data is taken from the social media of the created location networks and are integrated with an algorithm to detect any form of outbreak of epidemic diseases. This approach can also be used to forecast the breakout of any form of epidemic diseases (Thapen, et al., 2016). Another approach will be to make use of Twitter API to extract the tweets with the epidemic name. Then the tweets are filtered based on a given criteria such as tweeted by patients or GP, with the help of support vector machine identifier (SVM) classifier (Aramaki, 2011). In fact, there are multiple NLP techniques that can be used to extract the tweet data based on the keywords and detect the outbreak of any form of epidemics. There are existing researches which demonstrate that the conventional sentiment

analysis methodologies can be successfully used for sentiment analysis in the social networks. This has been in practice since the early 2000. There are multiple evolutions of various types of sources where opinions can be voices. Hence, the current opinion methodologies might no longer be effective, in this redeveloped environment. In this environment, multitude of issues need to be derived from the conventional sentiment analysis along with natural language processing. Overall, this creates a challenging environment with different set of complexities that includes, noisy content, short messages, variant form of metadata (age, sex, location). It is a known fact, that social networks create a clear impact on the languages, and this has become a core challenge of sentiment analysis. There is a constant evolution of language on the social network, which is used to generate the online content. Also, most of the written languages is visualized though some electronic screens such as desktop, laptop, tablets or phones, hence it could be said that the interaction partly happens with the help of technology. Moreover, the language that is used on the social media is more of malleable nature in comparison to the language that has been used for formal writing. The social media language is made up of personal communication and informal opinions, which is afforded by the mass users of the social media platform. This actually makes it more difficult for the conventional sentiment analysis method to analysis the inherent opinion from the given text. Hence, in order to adapt the changing language structures, research needs to implement strong natural language processing skills and linguistic skills, along with the conventional methodologies of sentiment analysis.

In this work we have taken the tweet data to arrive at the prediction of epidemic hit areas or the probability of being affected by any major epidemic that can harm the lives and property. We have used machine learning approach to arrive at the prediction and for comparison and analysis we have used different feature extraction techniques and algorithms to select the best out of it. In the next section we will be discussing about the dataset that we have tried to generate out of the tweet data from the twitter and transforming it into the structured from un-structured data and making it a *supervised learning* problem. After that we will discuss about the structure of the system for predicting the epidemic and different methodologies taken up for arriving at the better result which in case is the *Accuracy* and *different parameters of the Confusion Matrix*. Following that will be the results section and Future aspects of the work and then conclusion.

2. DATA DESCRIPTION AND DATA PROCESSING

The work here, Epidemic Search model using the power of the social network data analysis and then using this data to provide a probability score of the spread and to analyse the areas globally going to suffer from any epidemic spread-out. The easily available social network data from Twitter which in other words known as the tweet-data is very helpful in providing a lot of information about any events happening globally.

2.1 Data Source

In the recent years, social networking has attracted a lot of users. Social networking sites like Facebook, Twitter, Instagram etc. creates a lot of data every second and a lot of information from that can be got. Hence, this creates a space for doing some challenging research by computationally analysing the sentiments and opinions of the textual data which are unstructured in their behaviour. To achieve this, a gradual practice has grown for extracting the information from the data available in the social networking sites like predicting the epidemic in this case. The accuracy of the predicting model thus can be found out from the modelling output. To arrive at the output of the scenario presented here, tweet data is analysed and to extract the tweet data "Twitter API" is used. API needs to be signed up on the twitter and also has to have a login into the developer Twitter account. Following it, an application or an API needs to be developed

which can be then used to provide the keys and the tokens for using it in the programming environment.

2.1.1 Data Extraction

The Twitter API can then be used with the Python Programming language to extract the tweets from the Twitter and store in a HDFS (Hadoop Distributed File System) which is a distributed file system that is designed to run on any commodity hardware.***Tweepy*** is a python library that can be used to extract the tweeter tweets. The tweets can be easily collected and can be stored in the JSON format. JSON is a syntax for storing as well as exchanging the stored data.

2.1.2 Database Management

As in present scenario we have large storage of tweets, storing it on a single system and analysis can be difficult due to large data. This problem can be solved using distributed system. Example for storage we can use HDFS file systems or Apache Cassandra database management system. Spark is a cluster-computing framework which can be used with them and for python we can use python supported spark system which is pyspark. This has the advantage of storing very large dataset and to be accessed reliably depending on the bandwidth of the user. Another advantage is in the distributed system many clusters can host and execute directly attach storage and user application tasks. In this system either MongoDB or Spark system can be directly used with Python to extract the tweets and store in the distributed clusters. MongDB is a free and open source cross platform document-oriented database program. In this json like documents are used which has schema. It works on concept of collection and document. Where a document is a set of key-value pairs. And Collection is a group of these MongoDB documents. Here, Collection is equivalent to a RDBMS table. Also, it is contained in a database which is a physical container for collections and each database gets its own set of files on the file system.

2.1.3 Pre-processing

Removing the stop words like the, an, a etc can be a good step as they don't determine the polarity or sentiment of the tweet. For this we mostly use stop words in English from nltk package in python. Removing hyper-links, citations, references, hash-tags, multiple white-spaces can be done by regular expressions which makes the tweet description free from the "unrelated" English words and "chat language".

2.1.4 Polarity Generation

The predicting variable or the dependent variable which in this case is the polarity of the tweet that is found out from the sentiment of the tweets using the text blob in python. It targets some commonly areas like POS tagging (Parts of Speech tagging), Noun-Phrase terms extraction from text, Sentiment Analysis, Text Classification, Language Translation in text etc. Here a simple function for doing such task is used as below which targets for the tweet sentiment analysis: The function for getting the tweet sentiment is as below, this is used to generate the polarity class for

```
def get_tweet_sentiment(tweet):
        '''
        Utility function to classify sentiment of passed tweet
        using textblob's sentiment method
        '''
        # create TextBlob object of passed tweet text
        analysis = TextBlob(clean_tweet(tweet))
        # set sentiment
        if analysis.sentiment.polarity > 0:
            return 'positive'
        elif analysis.sentiment.polarity == 0:
            return 'neutral'
        else:
            return 'negative'
```

Figure 1: Polarity Generation Function

the tweet. Thus, making the unstructured data into a structured data. Below is the head of the data:

	Polarity	Text
0	positive	RT @GrantBrooke: On Cholera: @TwigaFoods will ...
1	positive	Good! Now SAVE THE BEES! https://t.co/x7qMctrlSg
2	positive	Jubilee gvt shuts Jacaranda & San Valencia...
3	negative	So what?... Does it make less of cholera https...
4	neutral	RT @washingtonpost: Scientists plan to trick Z...

Figure 2: Head of the Data

2.1.5 Hash Tag Analysis

All the words starting with the symbol # are hash-tags. These are helpful in understanding the trending issues. A word-cloud is an image representing the text in which the size of each word is proportional to the frequency of occurrence. Hence the bigger words are the most tweeted topics. A glance at the word cloud shows that most tweets are about the social problems like diseases, malnutrition, starvation and some countries affected by them. In our case, most frequent hash tagged words are Yemen, Cholera, Cholera Nairobi, zika, vaccines, The Story Of Yemen etc. Below is the function that is used for the hash tag analysis.

```
# Get all the hashhtag words that has "#"
hashtags = ""
for line in tweets:
    words = line.split()
    for w in words:
        if w.startswith("#"):
            hashtags += w + " "
```

Figure 3: Hash Tag Function

When the generated hash tags are generated and represented as the word cloud, it looks as below:

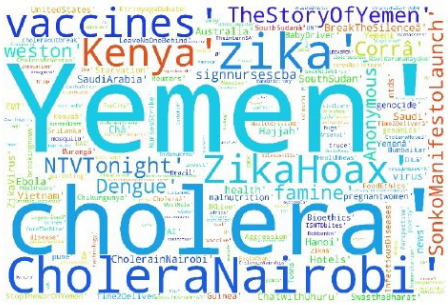

Figure 4: Hash-Tag word Cloud

2.1.6 Top-Users/Citations

Similar to analysing hashtags, we can extract the usernames usually preceded by @ symbol. The word cloud for the Citations is as below figure.

Figure 5: Citations Word Cloud

From the above word-cloud, we find that the top cited words are "washingtonpost", "Waabui", "GrantBrooke","TwigaFoods,"ICRC" etc.

3. SYSTEM DESIGN

In the system design section, here it will be present how the steps are followed to arrive at the prediction results.

1. Step: Tweet Data Streaming- Using Tweeter API

2. Step: HDFS MongoDB- Tweets extracted stored in MongoDB using Python library *pymongo*

3. Step: Dataset for Training-Available data as text data from tweeter is highly unstructured and noisy in nature and to use it for the modelling purpose, it needs to be cleaned. The different pre-processing techniques used as follows:

   (a) Escaping HTML characters
   (b) Decoding data
   (c) Removal of Stop-words

(d) Removal of Punctuation
(e) Removal of Expressions
(f) Split Attached Words
(g) Removal of URLs
(h) Removal of quotes
(i) Removing tickers
(j) Removing line-break, tab and return
(k) Remove whitespaces

2. Step: Label or Polarity Generation-The generated tweets without the sentiments class, imputed with the class using the *TextBlob* library in Python. Three classes are generated: *Positive*, *Neutral* and *Negative.*

3. Step: Feature Extraction- Different techniques used to analyse the accuracy of the prediction:

   (a) Bag of Words using CountVectorization, Uni-Grams and Bi-Grams
   (b) TF-IDF - Creating a unique value for the terms in a particular document.
   (c) Topic Modelling using LDA - To generate the topics for the corpus

4. Step: Machine Learning Model-The Features extracted or generated is fed into the Machine Learning Model/Algorithms to generate the results.

5. Step: Results Generation

Below is the flow chart Graph for the above Algorithm:

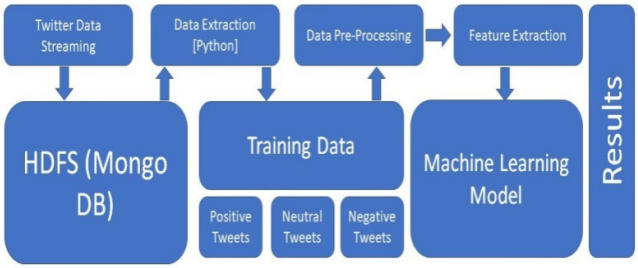

Figure 6: System Design Flow Chat of the Epidemic Prediction from Tweets

4. EXPERIMENTS AND RESULTS

As the unstructured data is converted into a supervised learning process, it is important to see the distribution and the counts of the different classes in the dataset. Let us now see how the distribution of the different classes belonging to the tweets is:

- *Positive Tweets*: Considering only the positive labelled tweets and extracting words, we can count the frequent words used in positive tweets. Word-cloud of positive tweets shows that they include health, water, vaccine, sanitation among other things. The *textblob* shows that there are 771 cases as termed as the positive class which is around 26.67% of the total tweets cases. The word-cloud for this classes is as below in the figure.

Figure 7: Positive Sentiment Word Cloud

- *Negative tweets*: Similarly, negative word-cloud shows that outbreak, dengue, worst, Yemen etc. are most used in negative tweets. For this case, the total number for the negative class tends to 692 which comprises of 23.96% of the total cases. The word-cloud for the negative classes is as below in the figure.

Figure 8: Negative Sentiments Word Cloud

- *Neutral tweets*: Word-cloud shows that the most used words in neutral tweets are hotel, cholera, Weston spread etc. The total for the neutral case tends to 1425 which is 49.34% of the total cases. The word cloud for this class is as below in the figure.

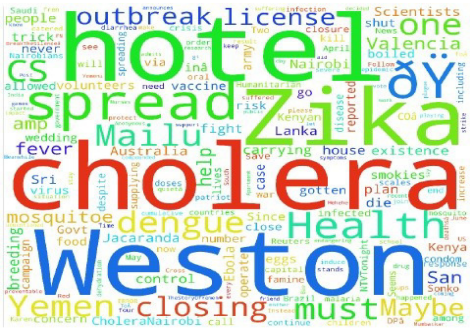

Figure 9: Neutral Sentiment Word Cloud

The histogram of the polarity of all the tweets from the *blob.sentiment polarity* shows the distribution of the polarity scores of each tweet class and is represented through the histogram as below:

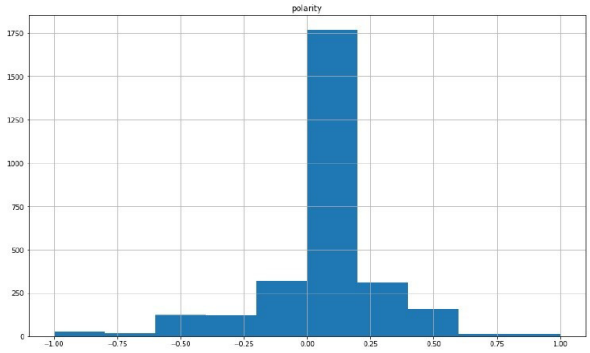

Figure 10: Histogram of the sentiment Polarity

The histogram shows that maximum polarity of the tweets lies in the range [0.0–0.24) approximately. If we closely observe the distribution that the tweet polarity follows tends to be "Normal Distribution". To analyse more on the length of the tweet and the polarity class of the tweet, when plotted, the histogram looks as shown below: This analysis shows that for the negative class, the frequency of the word counts is mostly more near to the 140-word count. The same is seen for the positive class but the frequency distribution is less than that of the negative class. For the neutral class, the frequency distribution is more in between the word counts around 130-150.

4.1.1Machine Learning

Once we clean the data and get a rough idea about the data, we can use any supervised ML model for sentiment classification since we already have the labels. Most of the approaches involving text classification uses *n-gram features*. This comes under Bag of words model as it doesn't care in the exact ordering of the words. Recent advanced models using RNN/LSTM models take the word order into consideration while classifying.

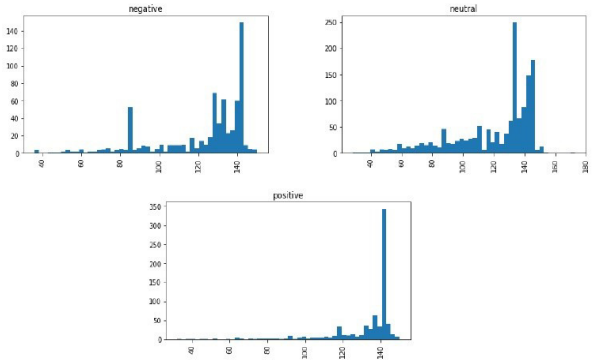

Figure 11: Histogram of the Sentiment Scores of different class

4.1.2 Sentiment Classifier:

Now for classifying the tweets and see how the prediction happens using different classifier, we use the above tweet dataset and pass it through any machine learning algorithm and see how the result is. Below is the comparison of the accuracies of the different models/algorithms that we have used: The above analysis shows that Decision tree classifier performed better followed by

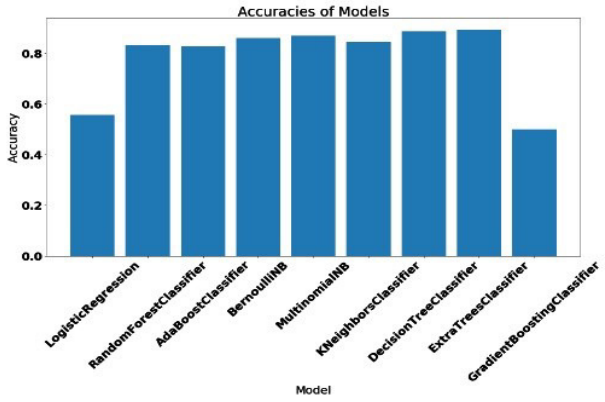

Figure 12: Comparison of the different algorithm performance

the K Nearest Neighbour Classifier.

4.1.3 Words n-Grams:

A tweet (such as a sentence or a document) is represented as the bag (multi set) of its words, disregarding grammar and even word order but keeping multiplicity.

- Strengths: Traditional, pretty solid feature representation.
- Weaknesses: Lose grammar/word order.
- Hyperparameters: The algorithm definition to execute

Below is the Feature Evaluation of different algorithms using the n-grams concept:

Naïve Bayes using Uni-grams and Bi-grams

Here we have used two Naïve Bayes algorithm. One is Bernoulli Naïve Bayes and other is Multinomial Naïve Bayes. Bernoulli Naïve Bayes follows Bernoulli distribution whereas Multinomial Naïve Bayes model is mainly based on the frequency of data. We have used the 5-fold cross-validation technique in which 4 out of 5 folds (in other words samples) is used for the training of the model and 1 out of 5 folds or the last fold is used for the validation of the model performance. From the 5-fold cross validation technique, average accuracy for the multinomial NB is found out to be 78.11% and that for the Bernoulli NB it's found to be 78.42%, minimum value for accuracy for multinomial NB is 61.53% and for Bernoulli NB is 65.16% and maximum value of accuracy for multinomial NB is 83.36% and for Bernoulli NB it's found to be 82.7%. We can find that on the basis of the average and minimum accuracy values, Bernoulli Naïve Bayes performed better than Multinomial Naïve Bayes. Whereas we find that the maximum accuracy value if for Multinomial Naïve Bayes.

For the hold-out Validation method, accuracy of multinomial NB is 84.43% and the confusion matrix for the same is as below:

Table 1: Confusion Matrix of Multinomial NB

Label	Predicted Negative	Neutral	Positive
Actual Negative	175	20	6
Actual Neutral	32	373	23
Actual Positive	22	32	184

From this we can analyse that out of 213 negative sample 175 is predicted correctly while 20 are wrong predicted to neutral class and 6 are predicted to positive class. Out of 423 neutral tweets only 32 are predicted to negative class and 23 are predicted to positive class which shows that our neutral labelled tweets are biased towards negative tweet. For positive sentiment tweets it gave total 54 wrong classification where 22 are negative classified and 32 are classified to neutral class and 184 are correctly classified. This also shows that chances of negative tweet to be predicted as positive are comparatively very less than positive tweet to be predicted as negative. Although it gave accuracy of 0.84 but misclassification is more when we see confusion metric. Same analysis we can see from precision, recall and f1-score values as shown below:

Table 2: Analysis

Class-name	Precision	Recall	F1-Score	Support
Negative	0.76	0.87	0.81	201
Neutral	0.88	0.87	0.87	428
Positive	0.86	0.77	0.82	238
Avg. / Total	0.85	0.84	0.84	867

Linear SVM using Uni-gram and Bi-grams

Accuracy for this model is 83.50% and when running using cross-validation method with 5-folds our accuracy ranges from 71.06% to 84.4%. The mean accuracy is around 80.09%. Hence, for the better representation and comparison we will be considering the mean accuracy for our final evaluation. Confusion metric for the model is:

Table 3: Confusion Matrix of SVM

Label	Predicted Negative	Neutral	Positive
Actual Negative	145	68	0
Actual Neutral	1	421	1
Actual Positive	0	73	158

From this we can analyse that out of 213 negative samples, 145 is predicted correctly while all wrong prediction is in neutral. This shows that negative and positive sentiments tweets can be separated much more easily as compared to negative and neutral sentiment tweets. Out of 423 neutral tweets only 2 tweets are predicted wrong and out of the remaining, 421 tweets are predicted correctly. This implies that the model performed better for the cases of neutral tweets. Also, for the positive sentiment tweets, it predicted 73 wrong classification out of 231. Which shows same pattern as that of the negative tweets.

If we see precision, recall and f1-score we can see that F1 score of neutral tweets are higher than negative and positive tweets which supports our analysis presented above.

Table 4: Analysis

Class-name	Precision	Recall	F1-Score	Support
Negative	0.99	0.68	0.81	213
Neutral	0.75	1.00	0.85	423
Positive	0.99	0.68	0.81	231
Avg. / Total	0.87	0.84	0.83	867

Thus, comparing the above analysis:

Table 5: Comparison Results of Different Algorithms

Algorithm	Value 1	Value 2	Value 3
Multinomial NB	78.11%	61.53%	83.36%
Bernoulli NB	78.42%	65.16%	82.70%
SVC	80.09%	71.06%	84.4%

SVC using "Linear" kernel performed extremely well for this analysis.

4.1.3 TF-IDF Vectorizer

We have used the scikit-learn's TFidfTransformer to arrive at the features to be input to the different classifier. We have used the following classifiers along with their performances:

- Naive Bayes: 94.145%
- SVC: 49.34%
- SVM(TFIDF): 87.9%
- Naive Bayes(TFIDF): 83.21%

Now including the n-gram analysis to our model, the following things we have included and built in the model:

- Unigram classifier (with mark-negation and without)
- Bigram classifier (with mark-negation and without)
- Unigram and bigram classifier (with mark-negation and without)

The following the result analysis:

- Unigram Classifier: 88.75%, 89.44%
- Bigram Classifier: 88.58%, 88.58%
- Unigram and Bigram Classifier: 89.10%, 88.75%

4.1.4 LSTM Networks

LSTM is a variant of recurrent neural network, which takes information of its previous time steps. In LSTM to handle drawback of Basic RNN cell for learning long sequences we corporates gates. For this model basic cleaning has been done and after that we have tokenized input tweets. After tokenization sentence is mapped with word index of its vocabulary and index of padding is kept as 0, padding ensure every input record have same length.

For passing sentences to model few points has been considered

- Preventing learning of least frequent words: vocabulary contains 5000 most frequent words.
- Each sentence is fixed with length of 500: smaller sentences are padded and longer sentences are truncated to length 500 words.

Here we are using self embedding technique for learning word embedding and embedded word size is kept at 32. The model consists of 5 layers:

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 500, 32)	160000
dropout_1 (Dropout)	(None, 500, 32)	0
lstm_1 (LSTM)	(None, 100)	53200
dropout_2 (Dropout)	(None, 100)	0
dense_1 (Dense)	(None, 1)	101

Figure 13: Architecture of LSTM

Below is the training of the LSTM model: We can see that the validation accuracy attained by LSTM is 67.04%.

```
Train on 1732 samples, validate on 1156 samples
Epoch 1/9
1732/1732 [==============================] - 158s - loss: -0.3619 -
acc: 0.5456 - val_loss: -0.8101 - val_acc: 0.6159
Epoch 2/9
1732/1732 [==============================] - 146s - loss: -2.0605 -
acc: 0.6443 - val_loss: -2.3857 - val_acc: 0.6237
Epoch 3/9
1732/1732 [==============================] - 152s - loss: -3.0746 -
acc: 0.6796 - val_loss: -2.7091 - val_acc: 0.6427
Epoch 4/9
1732/1732 [==============================] - 167s - loss: -3.3368 -
acc: 0.7021 - val_loss: -2.8678 - val_acc: 0.6514
Epoch 5/9
1732/1732 [==============================] - 152s - loss: -3.4819 -
acc: 0.7252 - val_loss: -2.9131 - val_acc: 0.6557
Epoch 6/9
1732/1732 [==============================] - 151s - loss: -3.5609 -
acc: 0.7396 - val_loss: -2.9819 - val_acc: 0.6644
Epoch 8/9
1732/1732 [==============================] - 160s - loss: -3.6095 -
acc: 0.7436 - val_loss: -2.8963 - val_acc: 0.6678
Epoch 9/9
1732/1732 [==============================] - 141s - loss: -3.6211 -
acc: 0.7494 - val_loss: -3.0327 - val_acc: 0.6704

Out[ ]:

<keras.callbacks.History at 0x2102682afd0>
```

Figure 14: Training results of LSTM

5. REPRESENTATION

For Visualizing there are a lot of tools available to showcase how the epidemic distribution is globally. Tableau is a visualization tool which can be connected to any database and different kind of visualization can be created to understand the data and better representation of the data.

In this project, Tableau is used to generate a visualization of the epidemic hit areas globally. A sample is presented below:

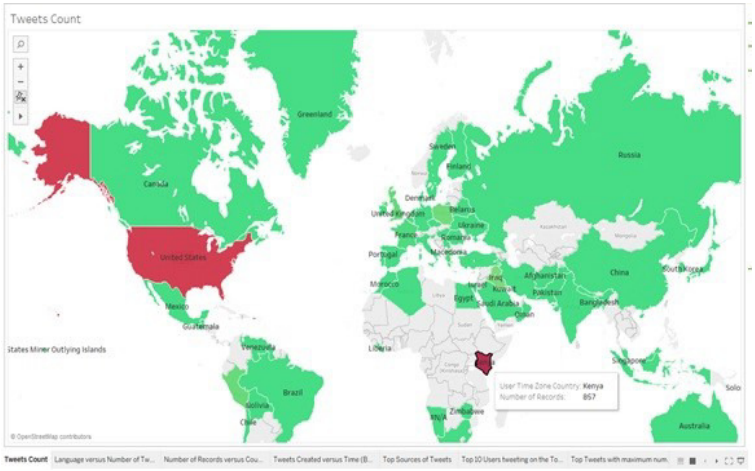

Figure 15: Epidemic Hit Regions using Tweet Analysis

6. CONCLUSION

The epidemic hit area prediction can be used to save a lot of lives globally. Here a lot of pre-processing of the unstructured data is done to make it to a structured data and different features extraction techniques like Count Vectorization, TF-IDF, Topic Modelling etc. is used to feed the data into the machine learning algorithms. The most important is the sentiment of the tweets to see how the polarity of the tweet is. The metric which is used here is the "accuracy" of the prediction and we have used the confusion matrix to arrive at the best performing algorithm. Naive Bayes using TF-IDF performed better than other methodologies and gave a better result.

REFERENCES

[1] Thomas, David R. "A general inductive approach for analyzing qualitative evaluation data." American journal of evaluation 27.2 (2006): 237-246.

[2] Pang, Bo, and Lillian Lee. "Opinion mining and sentiment analysis." Foundations and Trends R in Information Retrieval 2.1–2 (2008): 1-135.

[3] Adhikari, Nimai Chand Das. "PREVENTION OF HEART PROBLEM USING ARTIFICIAL INTELLIGENCE."

[4] Waaijenborg, Sandra, et al. "Waning of maternal antibodies against measles, mumps, rubella, and varicella in communities with contrasting vaccination coverage." The Journal of infectious diseases 208.1 (2013): 10-16.

[5] Miner, Gary, John Elder IV, and Thomas Hill. Practical text mining and statistical analysis for non-structured text data applications. Academic Press, 2012.

[6] Barbosa, Luciano, and Junlan Feng. "Robust sentiment detection on twitter from biased and noisy data." Proceedings of the 23rd international conference on computational linguistics: posters. Association for Computational Linguistics, 2010.

[7] Han, Eui-Hong Sam, George Karypis, and Vipin Kumar. "Text categorization using weight adjusted k-nearest neighbor classification." Pacific-asia conference on knowledge discovery and data mining. Springer, Berlin, Heidelberg, 2001.

[8] Pereira, Fernando C., Yoram Singer, and Naftali Tishby. "Beyond word n-grams." Natural Language Processing Using Very Large Corpora. Springer, Dordrecht, 1999. 121-136.

[9] Niesler, Thomas R., and Philip C. Woodland. "A variable-length category-based n-gram language model." Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on. Vol. 1. IEEE, 1996.

[10] Adhikari, Nimai Chand Das, Arpana Alka, and Raju K. George. "TFFN: Two Hidden Layer Feed Forward Network using the randomness of Extreme Learning Machine."

[11] Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. "Thumbs up?: sentiment classification using machine learning techniques." Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. Association for Computational Linguistics, 2002. [12] Dave, Kushal, Steve Lawrence, and David M. Pennock. "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews." Proceedings of the 12th international conference on World Wide Web. ACM, 2003.

[13] Joachims, Thorsten. A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. No. CMU-CS-96-118. Carnegie-mellon univ pittsburgh pa dept of computer science, 1996.

[14] Tang, Duyu, Bing Qin, and Ting Liu. "Document modeling with gated recurrent neural network for sentiment classification." Proceedings of the 2015 conference on empirical methods in natural language processing. 2015.

[15] Adhikari, Nimai Chand Das, Arpana Alka, and Rajat Garg. "HPPS: HEART PROBLEM PREDICTION SYSTEM USING MACHINE LEARNING."

Authors

Nimai Chand Das Adhikari received his Master's in Machine Learning and Computing from Indian Institute of Space Science and Technology, Thiruvananthapuram in the year 2016 and did his Bachelor's in Electrical Engineering from College of Engineering and Technology in the year 2011. He is currently working as a Data Scientist for AIG. He is a vivid researcher and his research interest areas include computer vision, health care and deep learning. He has started the Analytic Labs research group.

Vamshi Kumar Kurva received his Master's in Machine Learning and Computing from Indian Institute of Space Science and Technology, Thiruvananthapuram in the year 2017. He is currently working as a Data Science Engineer for FireEye. His interest areas include deep learning, video analytics, medical application and NLP.

Sankalp Kumar Nayak has 9+ years of experience in SAP Data Analytics. He has handled and worked on multiple projects related to SAP data analytics in all of its phases (Implementation, support/maintenance, up-gradation and roll-out). Also has 2+ years of experience in dealing in RPA(Automation) projects which is also referred as Business Process Automation.

Ashish Kumar Nayak received his Post Graduate diploma in Applied Statistics from Indira Gandhi Open University in the year 2017 and did his Bachelor's in computer science engineering from Konark Institute of Science and Technology in the year 2010. He is currently working as a Data scientist for Accenture in finance domain. His interest area includes Machine Learning, Computer Vision, NLP and statistical analysis.

Suhas S received his Master's in Machine Learning and Computing from Indian Institute of Space Science and Technology, Thiruvananthapuram in the year 2016 and did his Bachelor's in

Electronics and Communication Engineering from NMIT, Bangalore in the year 2014. He is currently working as a Research Associate in Indian Institute of Science(IISc). He is a 'Big data' enthusiastic individual with vested interest in 'Data Science' and his research interest areas include predictive modelling, computer vision and applied deep learning.

Kumar Rishav is pursuing dual degree course from Indian Institute of Space Science and Technology. He is currently pursuing Optical Engineering (Masters) and has completed B.Tech in Engineering Physics. He is doing his final year master thesis project from Institute for Applied Optics, University of Stuttgart. He is also a part of the Analyticlabs Research Group and leads the web development team. His interest areas are NLP, Computer Vision, Optics and Physics and Web Development with API

Vaisakh Shaj received his Master's in Machine Learning and Computing from Indian Institute of Space Science and Technology, Thiruvananthapuram in the year 2016. He is currently working as a Data Scientist for Intel. He is a vivid researcher and his research interest areas include computer vision, health care and deep learning.

Arpana Alka received her Master's in Machine Learning and Computing from Indian Institute of Space Science and Technology, Thiruvananthapuram in the year 2017 and did her Bachelor's in Computer Science Engineering from National Institute of Technology, Surat in the year 2014. She is currently working as a Data Science Engineer for Busigence Technologies. Her interest areas include deep learning, video analytics, medical application and NLP.

Hitesh Nayak received his Master of Business Administration from Great Lakes Institute of Management, Chennai in the year 2016 and did his Bachelor's from National Institute of Science and Technology, Odisha in the year 2011. He is currently working as a Data Scientist in Prescience Decision Solutions and has 5 years of experience. His interest areas are Forecasting, Deep Learning, Business Analysis, Management and NLP.

Karthikeyan received his Master's in Software Engineering from VIT University, Vellore in the year 2017. He worked as a Frontend Associate and Fullstack Engineer in Busigence technologies, Bengaluru. He has an in-depth specialization in developing business logic for the backend. His interest areas include various kinds of web technologies and API designs.

Srikant Nayak is persuing his Master's in machine learning and computing from Indian institute of space science and technology,thiruvananthapuram and did bachelor's in electrical engineering from institute of technical education and research in the year 2012. His research areas are computer vision, machine learning, deep learning and NLP.