Modelling Ukraine-Russia Conflict through Sentimental Analysis

Aniruddha Redij
Electrical Engineering
Stevens Institute of Technology
Hoboken, USA
aredij@stevens.edu

Pranavi Vashishtha
Applied Artificial Intelligence
Stevens Institute of Technology
Hoboken, USA
pvashish@stevens.edu

Shreeya Kokate
Machine Learning
Stevens Institute of Technology
Hoboken, USA
skokate@stevens.edu

***Abstract*—Twitter nowadays is considered as a reservoir of raw information and hence it explains its efficacious nature in research related to natural language processing. Sentimental analysis being a crucial part of natural language processing is a powerful tool to access the raw opinion of people on a world-wide platform. Hence, this potent technique is used to access the ongoing verbal war on Twitter related to the Ukraine Russia conflict which is triggering emotion in public on political debate topics such as, 'Humanitarian Corridor', 'Communism', 'Finlandization', 'Terrorism', 'Nuclear technology', etc. The presented study is concerned with the investigation of three different algorithms which have been proposed for detecting the sentiment of people (classified into neutral, negative, and positive) on this sentiment analysis project. The presented project is concluded with the results obtained from this study which are related to the general categorization of variegated opinions of people.**

Keywords – Twitter, natural language processing, Sentimental analyses, Ukraine Russia conflict, political debate, Classification

I. INTRODUCTION

Twitter is an online news and social networking site where people communicate in short messages called 'tweets'. It is a popular microblogging site. Twitter has a lot of drivel, but at the same time, there is a base of useful news and knowledgeable content. However, a growing number of Twitter users send out useful content, and that's the real value of Twitter. The sentiment analysis strategy is used which is based on extracting trading signals using machine learning algorithms applied to social media data (Twitter in this case). The process started with the collection of most frequent tweets that contained at least one keyword listed in vocabulary over a predefined time frame. A total of ten thousand tweets filtered within a certain time frame of this conflict period were collected. This data is further processed by the machine learning algorithms aiming to extract models that are used to predict the result of the public sentiment. The aim is to understand the public sentiment on the ongoing conflict between Ukraine and Russia. We collected the data from twitter and formed the dataset featuring the sentiments of the critics on war between Russia and Ukraine. For data collection, we can use Tweepy API(Twitter's official data scraping api), Twint API, snsCRAPER(which we have used). We collected the data from twitter and formed the dataset featuring the sentiments of the critics regarding the current war situation between Russia and Ukraine. Thus, providing a brief overview of the problem to be addressed and highlighting the methodology by implementing machine learning techniques.

II. RELATED WORK

Sentiment analysis of tweets data is considered as a much harder problem than that of conventional text such as review documents. This is partly due to the short length of tweets, the frequent use of informal and irregular words, and the rapid evolution of language in Twitter. A large amount of work has been conducted in Twitter sentiment analysis following the feature-based approaches. One of the methods that has been previously implemented uses annotation method. This method is based on developing a corpus of Ukrainian and Russian news and then annotated each text with three categories: positive, negative and neutral. Each text was marked by at least three independent annotators via the web interface and the texts marked by all three annotators with the same category were used in the further experiments. Other methods include just mining of the some important keywords from the twitter and then being limited to the analyses part without implementing any machine learning algorithms to evaluate the data. Other methods use sentiment analyses but are implemented on a different targeted problem. Several studies have been conducted regarding sentimental analysis and Twitter tweets on various domains, some of these domains include health care (Gohil, Vuik, Darzi, 2018), movie reviews (Jain, 2013) and others. Prior studies have been conducted using emotional analysis on Twitter data for example the work carried out by Mathur et al who look at a wide distribution of Twitter data and categorize these tweets into emotions. This analysis helps to understand the mental health of people on Twitter (Mathur, Kubde, Vaidya, 2020).

This study focuses on understanding public's perspective of current Ukraine and Russia's political conflict by applying sentimental analysis and studying the results of four different algorithms. Analyzing the results to understand the world-wide public opinion on the war(i.e.Positive,Negative,Neutral) Look at figure no. 1,2 and 3 for data visualisation.

III. OUR SOLUTION

A. *Description of Dataset*

The scrapping of 25,000 tweets in US English language was completed within the peak political conflict time span with all the information formulated in a data frame which included the date, the content of the tweet, user, hashtags, mentioned user, etc. as the features of the dataset. For this project the snscrape library is used. snscrape is a scraper for social networking services (SNS). It scrapes things like user profiles, hashtags, or searches and returns the discovered items, e.g. the relevant posts. Created a data frame which included these features of the dataset. Later, data cleaning is being be done by removing the unwanted columns, duplicate data values, URL, numbers, mentions, punctuation, stopwords and short words from the data frame to obtain a formulated and desired data frame with the information needed. They can safely be ignored without sacrificing the meaning of the sentence. For example, the words like the, he, have etc. In our case the words that we have considered eliminating are : ['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'yourself',. . . .]. Also, spelling correction was taken care of. Converting emoticons and emojis to words is also important to express these into computer understandable format. The next step is to pre-process the data. Once the data has been cleaned the nest step is tokenization. Tokenization is essentially splitting a phrase, sentence, paragraph, or an entire text document into smaller units, such as individual words or terms. Before processing a natural language, we need to identify the words that constitute a string of characters.Stemming is the process of reducing a word to its word stem that affixes to suffixes and prefixes or to the roots of words known as a lemma. Stemming is a technique used to extract the base form of the words by removing affixes from them. It is just like cutting down the branches of a tree to its stems. For example, the stem of the words eating, eats, eaten is eat. Lemmatization usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma. Lemmatization is extremely important because it is far more accurate than stemming. Part-of-speech (POS) tagging is a process which refers to categorizing words in a text (corpus) in correspondence with a particular part of speech, depending on the definition of the word and its context. A POS tag is a special label assigned to each token (word) in a text corpus to indicate the part of speech and often also other grammatical categories such as tense, number (plural/singular), case etc. POS tags are used in corpus searches and in text analysis tools and algorithms. This step is the most important in order to separate the relevant and most important data for machine learning models to consider it as the input resulting in a better accuracy than a dirty data collection. Data preprocessing is a required first step before any machine learning machinery can be applied, because the algorithms learn from the data and the learning outcome for problem solving heavily depends on the proper data needed to solve a particular problem – which are called features. We will to this through Lemmatization, Stemming, Tokenization and POS Tagging.

Data visualization will include how features is related to each other and plot relevant graphs to easily describe the dataset. We are further studying the extent of emotion categorization by tracking the number of tweets for each category. we choose to create our own dataset instead of using a pre-existing datasetWe develop a full understanding of the problem and our data. We can capture all the edge cases that automated systems might miss or restrict us from considering, flexibility or the ease of customization in the data, data quality assurance and achieving high accuracy and ensure data security.

Following are the statistcs of our dataset being created and further used:

- The size of the dataset : (25000, 27)
- Over the period of - 15th February to 15th March 2022
- Average no. of words per tweet: 24.63
- Spread of no. of words used per tweet: 12.29
- Minimum number of words used per tweet: 1
- Maximum number of words used per tweet: 61

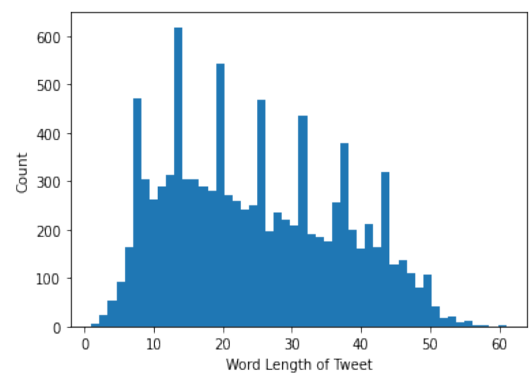

Fig 1. Word length of tweet vs count

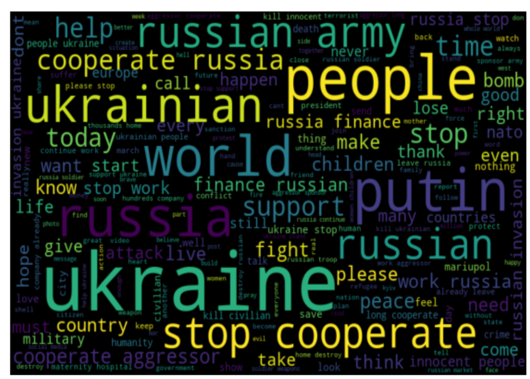

Fig 2. Word cloud plot of pre-processed data

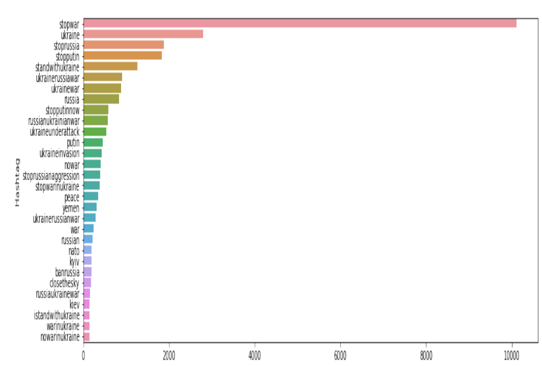

Fig 3. Hashtags and its counts

B. Implementation Details

1) : Exploratory Analysis

Sentiment Analysis

Sentiment analysis is a natural language processing (NLP) technique used to determine whether data is positive, negative or neutral. Sentiment analysis is often performed on textual data to help monitor sentiment in public and understand public views. It is used in process of detecting positive or negative sentiment in text. This method is often used by organizations to detect sentiment in social data to gauge brand reputation and understand the gravity of socio-political inferences . Since humans express their thoughts and feelings more openly than ever before, sentiment analysis is fast becoming an essential tool to monitor and understand sentiment in all types of data. Automatically analyzing feedback such as opinions in survey responses and social media conversations allows governments and organizations to learn what makes public happy or frustrated, so that they can tailor the marketing campaign that meet the public needs. Sentiment analysis is an approach to natural language processing (NLP) that identifies the emotional tone behind a body of text. It gives us measure of the sentiment of the text as positive, negative or neutral. Textblob can be used for complex analysis and working with textual data. When a sentence is passed into Textblob it gives two outputs, which are polarity and subjectivity. TextBlob calculates subjectivity by looking at the 'intensity'. Intensity determines if a word modifies the next word. As TextBlob is a Lexicon-based sentiment analyzer It has some predefined rules or we can say word and weight dictionary, where it has some scores that help to calculate a sentence's polarity. That's why the Lexicon-based sentiment analyzers are also called "Rule-based sentiment analyzers". One of the great things about TextBlob is that it allows the user to choose an algorithm for implementation of the high-level NLP tasks: PatternAnalyzer - a default classifier that is built on the pattern library. NaiveBayesAnalyzer - an NLTK model trained on a movie reviews corpus. When calculating a sentiment for a single word, TextBlob uses the "averaging" technique that is applied on values of polarity to compute a polarity score for a single word and hence similar operation applies to every single word and we get a combined polarity for longer texts. The polarity score is a float within the range [-1.0, 1.0]. The subjectivity is a float within the range [0.0, 1.0]

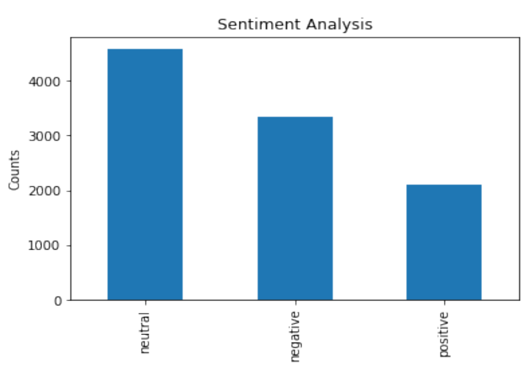

Fig. 4 Plotting the Positive, Negative and Neutral tweets from the pool of dataset

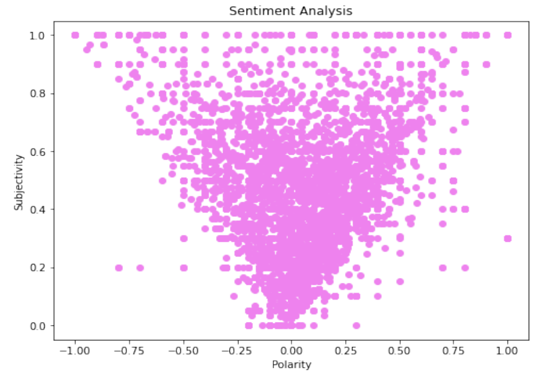

Fig. 5 Plotting the Subjectivity and Polarity for the sentiments of each tweet

Emotional Analysis

In some cases, the sentiment analysis might not enough understand the public view entirely. Emotion analysis is the process of identifying and analyzing the underlying emotions expressed in textual data. Emotion analysis, is a more involved, deeper analysis of emotions that tries to drill down into the psychology of the users or the writer. It is the technique of finding and interpreting the emotions conveyed in textual material which is more than just positive, negative and neutral.

Text2emotion works to extract the emotions from the text. Text2Emotion is the python package that will assist you to pull out the emotions from the content. Processes any textual data, recognizes the emotion embedded in it, and provides the output in the form of a dictionary. It is compatible with 5 different emotion categories as Happy, Angry, Sad, Surprise and Fear.

```
[14] data_set.head()
```

	Datetime	Tweet Id	Text	Username	clean_text	text_preprocessed	Emotion	Angry	Surprise	Sad	Fear
0	2022-03-14 23:59:24+00:00	1503521205873438721	russia-ukraine talks done for the day; eu agre...	PeterBrandt007	russiaukraine talks done day eu agrees fresh s...	russiaukraine talk agree fresh sanction	{'Happy': 0, 'Angry': 0, 'Surprise': 0, 'Sad':...	0.0	0.0	0.0	0.0
1	2022-03-14 23:58:28+00:00	1503520971822706692	it's really heartbreaking! 🙁 rest in peace! #s...	ahmadmarufS	really heartbreaking 🙁 rest peace	really heartbreaking rest peace	{'Happy': 0.0, 'Angry': 0.0, 'Surprise': 1.0, ...	0.0	1.0	0.0	0.0
2	2022-03-14 23:57:40+00:00	1503520771494539264	to russian people\nplease believe in her\nplea...	galaxys23172864	russian people please believe please stop war	russian people please believe please stop	{'Happy': 0.0, 'Angry': 0.0, 'Surprise': 0.0, ...	0.0	0.0	1.0	0.0
3	2022-03-14 23:51:55+00:00	1503519323943489550	@toddyspace i support you 👍 \n@nobelprize \nUAE...	ZOTILO2	support 👍 UAEA	support UAEA	{'Happy': 0, 'Angry': 0, 'Surprise': 0, 'Sad':...	0.0	0.0	0.0	0.0
4	2022-03-14 23:51:08+00:00	1503519125330661381	@innavoa i support you 👍 \n@nobelprize \nUAEA\n...	ZOTILO2	support 👍 UAEA	support UAEA	{'Happy': 0, 'Angry': 0, 'Surprise': 0, 'Sad':...	0.0	0.0	0.0	0.0

Fig. 6 Emotions of each tweets are identified by giving a score.

2) : Machine Learning Models

The presented study is using three machine learning algorithms on the analysed data after using sentimental analyses techniques for classification and categorisation purposes. The four machine learning algorithms that have been used are – 'Multinomial Naïve Bayes', 'Logistic Regression' and 'Linear Support Vector Classification'. All these algorithms have been chosen to categorize the data that has been generated after the application of Latent Dirichlet Allocation (LDA) Topic Modelling which is used to generate nine different topics, extracted from the base data. Since, the aim for this project is to understand public's perspective of current Ukraine and Russia's political conflict by applying sentimental analysis and studying the results of four different algorithms and further analysing the results to understand the world-wide public opinion on the war (i.e. Positive, Negative, Neutral), hence there is a need of algorithms which help us to classify the emotion.

1. LDA Topic Modelling:

Latent Dirichlet Allocation (LDA) is a popular topic modeling technique to extract topics from a given corpus. The term latent conveys something that exists but is not yet developed. In other words, latent means hidden or concealed. Now, the topics that we want to extract from the data are also "hidden topics". It is yet to be discovered. Hence, the term "latent" in LDA. The Dirichlet allocation is after the Dirichlet distribution and process. This process is a distribution over distributions, meaning that each draw from a Dirichlet process is itself a distribution. What this implies is that a Dirichlet process is a probability distribution wherein the range of this distribution is itself a set of probability distributions.

For the implementation process the packages that have been used from the sklearn library. The data on which these packages will work contains 10001 and 19 columns. For reference look at Fig no.7. After the completion of this step, splitting of the data, training and testing has been carried out. Once the refining and tokenization of the data has been completed, we vectorize the data followed by the generation of the matrices with the specified number of columns using the randomized algorithm. The outputs of the model include two matrices: one is the topic probability distributions over documents, represented by an N × K matrix; the other is the word probability distributions over topics, represented by a K × V matrix. We then fit and transform the model to categorize the data into 10 different topics which contain relevant tokens. In relation to our code, the final output is a compilation of 10 topic modelled into different lists of words. Some of the words also contain emojis to express emotion.

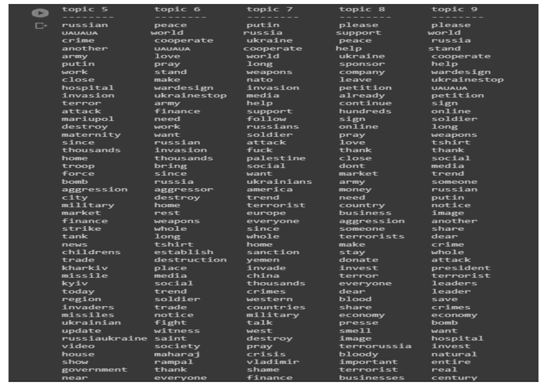

Fig. 7 Categorized words into relevant topics

2. Multinomial Naive Bayes:

The Multinomial Naive Bayes algorithm is a Bayesian learning approach popular in Natural Language Processing (NLP). The program guesses the tag of a text using the Bayes theorem. It calculates each tag's likelihood for a given sample and outputs the tag with the greatest chance. The multinomial Naive Bayes classifier is suitable for classification with discrete features.We use Naive Bayes Classifier as it is the simplest classification method and thus our base model. This have been used because of the following reasons:

- This algorithm can be used on discrete data; Since we have used only 20000 initial tweet values.
- It is easy for this algorithm to predict real time applications; since the twitter analyses is implemented on real time twitter data.
- It has the capability to easily handle large datasets and also because it is highly scalable.; Since the dataset that we have used is quite large based upon the tweets related to both side of Ukraine-Russia conflict.

For the implementation of the Na¨ıve Bayes method we are using the sklearn class from the scikit-learn library. Further splitting, training and testing of the data, the algorithms that has been applied is the Multinomial Naive Bayes algorithm. To start with its implementation, the trained and tested data has been fitted to the Naive Bayes classifier according to X, y for prediction purposes and then further the accuracy and confusion matrix has been calculated. Further, prediction method has also been used to Perform classification on an array of test vectors X. To apply the multinomial naïve bayes model, sklearn library has been used to import the model. This particular algorithm is being used because the multinomial Naive Bayes classifier is suitable for classification with discrete features (e.g., word counts for text classification). The multinomial distribution normally requires integer feature counts.

To use the algorithm on a good way, we have to transform the database as we did for CategoricalNB or in any other numerical way (Spoiler alert: it is perfectly adapted for counting features). Nyi represents the sum of the values for the class y and the feature i + the Laplace parameter. For the denominator, Ny represents the overall sum of all the values in the database for the class y + alpha to be multiplied by n which represents the number of features used (without the target). Nyi Ni are the sum of the values and not the length. Which means that the way we give a value to the category (like 0–1–2-… or 1–3–5-…) will have an impact on both the sums in the numerator and denominator which will impact the overall calculus and make it suboptimal. As a result, the Multinomial Naive Bayes is widely used for text classification where we have counting data like Bag of Words or TFIDF. Calculate all the parameters to get the likelihood for the required classes + the evidence and we can calculate the posterior probabilities. The most highest probability is given the preference and is being considered. We have obtained an accuracy of 80percent. Observe Figure 8 for reference purposes.

```
Accuracy of Naive Bayes is:  0.8018782187215995
Confusion Matrix:
 [[2601  651]
 [   3   46]]

[ ] print(classification_report(Y_test, y_pred))

              precision    recall  f1-score   support

           0       0.80      1.00      0.89      2604
           1       0.94      0.07      0.12       697

    accuracy                           0.80      3301
   macro avg       0.87      0.53      0.51      3301
weighted avg       0.83      0.80      0.73      3301
```

Fig. 8 Result of Naive Bayes

3. Logistic Regression:

Logistic regression is a simple yet very effective classification algorithm so it is commonly used for many classification tasks. It uses a logistic function to model the dependent variable. This has been used because of the following reasons:

- Logistic regression is easier to implement, interpret, and very efficient to train.
- It can easily extend to multiple classes (multinomial regression) and a natural probabilistic view of class predictions; since the LDA topic modelling is dividing the data into nine different topics.
- It is very fast at classifying unknown records; Since we need an efficient algorithm to classify the random topics that have been generated based on the labels.

For the implementation of the logistic regression method we are using the sklearn class from the scikit-learn library.

To start with its implementation, the trained and tested data has been fitted to the logistic regression classifier according to X, y for prediction purposes and then further the accuracy and confusion matrix has been calculated. Further, prediction method has also been used to Predict class labels for samples in X. To apply the logistic regression model, sklearn library has been used to import the model. Before we build the model, we use the standard scaler function to scale the values into a common range. Next, we create an instance of LogisticRegression() function for logistic regression.We are not passing any parameters to LogisticRegression() so it will assume default parameters. Some of the important parameters you should know are – penalty: Default = L2 – It specifies the norm for the penalty ;C: Default = 1.0 – It is the inverse of regularization strength ; solver: Default = 'lbfgs' – It denotes the optimizer algorithm. Here we are also making use of Pipeline to create the model to streamline standard scalar and model building. In the next step, we fit our model to the training data with the help of fit() function. We can further try to improve this model performance by hyperparameter tuning by changing the value of C or choosing other solvers available in LogisticRegression(). We have currently obtained an accuracy of 83 percent.

```
acc_lr = accuracy_score(y_pred1, Y_test)
print("Accuracy of LR is: ",acc_lr )
print("Confusion Matrix: \n", confusion_matrix(y_pred1, Y_test))

Accuracy of LR is:  0.8388367161466223
Confusion Matrix:
 [[2586  514]
 [  18  183]]

[ ] print(classification_report(Y_test, y_pred1))

              precision    recall  f1-score   support

           0       0.83      0.99      0.91      2604
           1       0.91      0.26      0.41       697

    accuracy                           0.84      3301
   macro avg       0.87      0.63      0.66      3301
weighted avg       0.85      0.84      0.80      3301
```

Fig. 9 Result metrics for Logistic Regression

4. Linear SVC:

This algorithm is used for linearly separable data, which means if a dataset can be classified by using a hyperplane, then such data is termed as linearly separable data, and classifier is used called as Linear SVC classifier. This has been used because of the following reasons:

- It is relatively better because it creates a clear margin of separation between classes.
- It Works in a way that it gives a best fit even if the data is heavy.
- It includes dense as well as sparse inputs.

For the implementation of linear SVC method we are using the sklearn class from the scikit-learn library.

In this model, the trained and tested data has been fitted to the Linear Support Vector Classifier according to X, y for prediction purposes and then further the accuracy and confusion matrix has been calculated. Further, prediction method has also been used to Predict class labels for samples in X. To apply the Linear Support Vector Classifier model, sklearn library has been used again. It is Linear Support Vector Classification. It is similar to SVC having kernel = 'linear'. The difference between them is that LinearSVC implemented in terms of liblinear while SVC is implemented in libsvm. That's the reason LinearSVC has more flexibility in the choice of penalties and loss functions. It also scales better to large number of samples. If we talk about its parameters and attributes then it does not support 'kernel' because it is assumed to be linear and it also lacks some of the attributes. However, it supports penalty and loss parameters as follows : penalty string, L1 or L2(default = ‘L2’). This parameter is used to specify the norm (L1 or L2) used in penalization (regularization) ; loss string, hinge, squaredhinge (default = squaredhinge) It represents the loss function where ‘hinge’ is the standard SVM loss and ‘squaredhinge’ is the square of hinge loss. Once fitted, the model can predict new values as well. Currently, we have achieved the highest accuracy with this algorithm as 88percentage.

```
Accuracy of Naive Bayes is:  0.8806422296273856
Confusion Matrix:
 [[2516  306]
 [  88  391]]

[ ] print(classification_report(Y_test, y_pred2))

              precision    recall  f1-score   support

           0       0.89      0.97      0.93      2604
           1       0.82      0.56      0.66       697

    accuracy                           0.88      3301
   macro avg       0.85      0.76      0.80      3301
weighted avg       0.88      0.88      0.87      3301
```

Fig. 10 Result metrics for Linear SVC

IV. Comparison

The primary objective of model comparison and selection is definitely better performance of the machine learning software/solution. The objective is to narrow down on the best algorithms that suit both the data and the business requirements. Model Selection and Evaluation is a hugely important procedure in the machine learning workflow. This is the section of our workflow in which we will analyse our model. We look at more insightful statistics of its performance and decide what actions to take in order to improve this model. We have calculated Precision, F1 Score, Recall and the respective accuracies as the evaluation metrics of our machine learning models.

By comparing accuracy output of all the three machine learning models, we observed that accuracy of the Linear Support Vector model is better than compared with rest other. The representation is given in form of bar graph as followed -

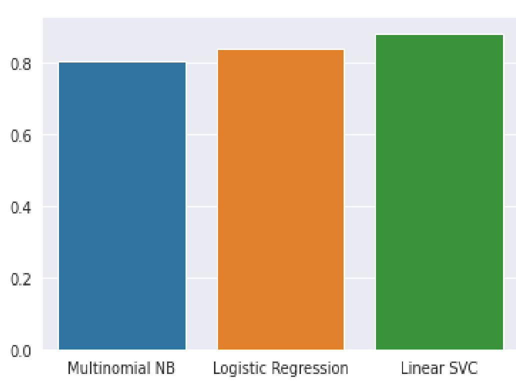

Fig. 11 Comparison of ML models

V. Conclusion

We have created our own dataset by Twitter data scrapping which was then cleaned and pre-processed in order to make it ready to use for machine learning operations. Further, we have implemented four machine learning concepts, namely, Topic Modelling, Logistic Regression, Linear SVC and Multinomial Naive Bayes algorithms. LDA does two tasks: it finds the topics from the corpus, and at the same time, assigns these topics to the document present within the same corpus. Multinomial Naive Bayes proves to be a good performer with an accuracy of 80 percentage. Logistic regression is easier to implement, interpret, and very efficient to train with an accuracy of 83 percentage. It is very fast at classifying unknown records Linear SVC model shows the highest accuracy so far among the models implemented with an accuracy of 88 percentage. We look forward to compare more classification model to explore the most accurate machine learning model which will help us learn the different categories of sentiments and emotions.

VI. References

- Gohil, S., Vuik, S., Darzi, A. (2018). Sentiment Analysis of Health Care Tweets: Review of the Methods Used. JMIR Public Health Surveill.
- Jain, V. (2013). Prediction of Movie Success using Sentiment Analysis of Tweets. The International Journal of Soft Computing and Software Engineering, 3(3), 308-313.
- Mathur, A., Kubde, P., Vaidya, S. (2020). Emotional Analysis using Twitter Data during Pandemic Situation: COVID-19. IEEE.
- Rodríguez-Ruiza, J., Mata-Sánchez, J. I., Monroy, R., Loyola-González, O. (2020). A one-class classification approach for bot detection on Twitter. Computers Security.https://ieeexplore.ieee.org/document/8100410
- https://thecleverprogrammer.com/2022/03/15/ukraine-russia-war-twitter-sentiment-analysis-using-python/
- https://www.csie.ntu.edu.tw/ cjlin/papers/linear-svr.pdf
- https://ieeexplore.ieee.org/document/8908018
- https://web.stanford.edu/ jurafsky/slp3/4.pdf
- https://arxiv.org/ftp/arxiv/papers/1601/1601.06971.pdf
- http://www.cs.columbia.edu/ julia/papers/Agarwaletal11.pdf
- https://link.springer.com/content/pdf/10.1007/978-3-642-35176-1-32.pdf
- https://towardsdatascience.com/topic-modeling-and-latent-dirichlet-allocation-in-python-9bf156893c24
- https://monkeylearn.com/sentiment-analysis/
- https://www.sciencedirect.com/topics/computer-science/emotion-analysis
- https://towardsdatascience.com/sentiment-analysis-concept-analysis-and-applications-6c94d6f58c17