Indian Institute of Technology Kanpur
Department of Mathematics and Statistics
MTH514A Course Project

Sentiment Analysis of Tweets during Russia-Ukraine War of 2022

Authors:
Abir Naha (201257)
Arkaprova Saha (201278)
Arkonil Dhar (201279)
Shreya Pramanik (201415)
Souvik Bhattacharyya (201433)

Supervisor:
Dr. Minerva Mukhopadhyay

April, 2022

Abstract

Since 2000's worldwide interconnection of individual networks aka internet started exploding to serve millions of users around the world. With evolving smartphone technology social media became a day-to-day phenomenon attracting users of different ethnicities and age groups. One of such social networking site is Twitter which started gaining popularity among rich and famous, in turn having a massive surge of users owing to their fanbase. Russia-Ukraine war to current date, is one of the widely discussed issue in Twitter. From these tweets using text mining techniques we can have an overview of people's opinion of the ongoing war. In our project we extracted a segment of Twitter data from the end of February to beginning of March and performed sentiment analysis. First we preprocessed the data using Tokenization, Lemmatization, then performed Feature extraction using Bag of Words, Tf-Idf and N-gram models. After that we applied dimension reduction technique and K-Means Clustering Algorithm to draw conclusions from our dataset.

Acknowledgement

It is a great pleasure for us to express respect and deep sense of gratitude to our supervisor Dr. Minerva Mukhopadhyay, Assistant Professor, Department of Mathematics and Statistics, Indian Institute of Technology Kanpur, as a convener of this course for her wisdom, vision, expertise, guidance, enthusiastic involvement and persistent encouragement during the planning and development of this work. We are truly grateful towards her for giving us a wonderful exposure to the topics of Multivariate Analysis.

Abir Naha
Arkaprova Saha
Arkonil Dhar
Shreya Pramanik
Souvik Bhattacharyya

Contents

1 INTRODUCTION

Twitter is a micro-blogging site where users interact with each other through messages known as 'tweets' unlike Facebook, or Whatsapp which mainly focuses on connecting with friends and family, in Twitter people talk about social, economic, political or similar issues. The data generated from Twitter gives us glimpse of social sentiments which can be used to discover emotion, opinion and attitude towards certain events through text mining processes. These opinions may be postive, negative, neutral in the sense that positive opinions are those which contains some good words for a certain news and negative opinions are those criticizing some event, neutral being the opinions in a middle ground. To draw insights from Twitter Sentiment data of ongoing **Ukraine-Russia Conflict 2022** we first have a quick look at sentiment analysis.

Sentiment Analysis

Sentiment analysis is contextually a natural language processing technique to identify a person's opinion through online conversations.With the advancement of artifical intelligence the ability of text mining has increased considerably. Sentiments can be positive, negative or neutral. There are many techniques such as feature extraction, tokenization, emotion study etc are applied to a natural text to determine the underlying sentiment. Here we would try to explain the approach we have taken in analyzing the sentiment of extracted Twitter data.

2 METHODOLOGY

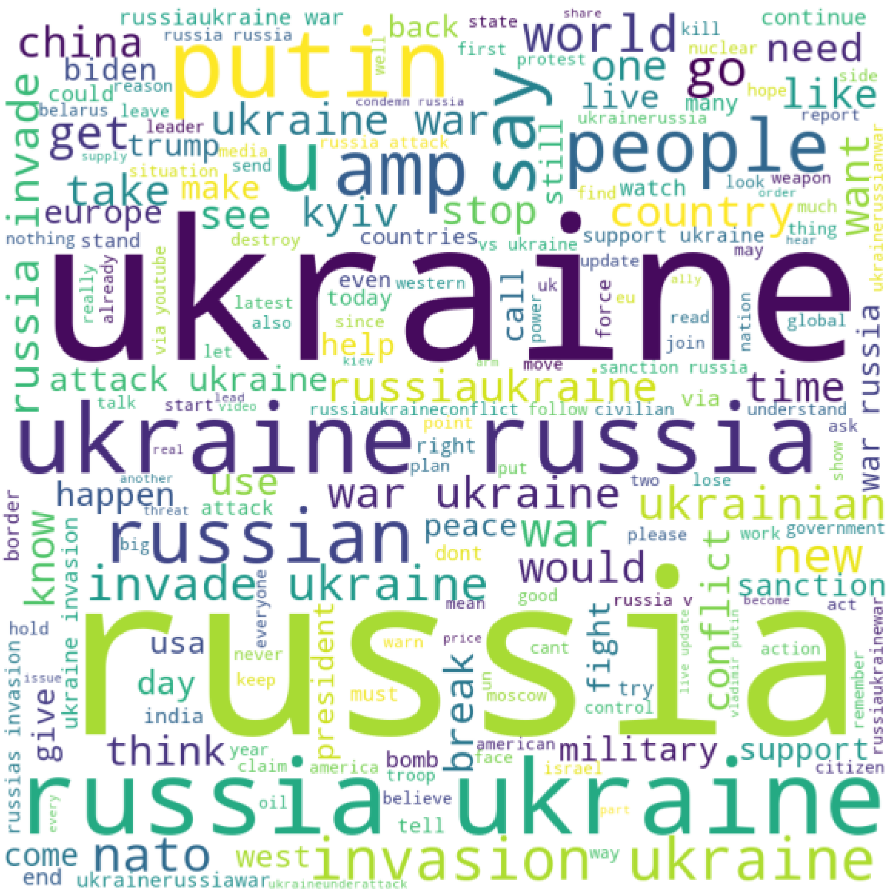
2.1 Data Extraction

We collected the data, using the python library `snscrape`, from 24th February, 2022 to 9th March 2022, and with keywords 'ukraine russia' present in it.

We collected the data seperately for each day and later merge them together. After that we had a total of 2,896,731 tweets to work with. Each single observation contained the following fields:

- **url**: the url of the tweets.
- **date**: date and time of posting the tweets.
- **content**: the actual content of the tweets.
- **tweet_id**: an unique id of each post.
- **user_id**: unique id of user who posted the tweet.

From these fields we used only the contents of the tweets for our analysis.



This is a wordcloud prepared using the most frequent words present in all of the tweets combined. The purpose of this visualization is to get a gross idea about the words that are appearing frequently in the tweets.

2.2 Preprocessing Text Data: Text Cleaning

Like most data, the Data collected from twitter are not ready to use for analysis and drawing insights. Most of these tweets contain username, empty spaces, special characters, stop words, abbreviations, hastags, timestamps, urls etc. So we performed several data cleaning processes as follows:

2.2.1 Lowercasing

Converting all the text data, into lower case helps in further preprocessing of the text. So we converted the texts into lowercase.

2.2.2 Removal of Unnecessary characters

Texts which are not significantly important for analysis, such as URLs, punctuation, extra spaces between, words, stopwords emoji, non-English words are removed.

Punctuation: Punctuation marks don't contribute significantly in meaning of the sentences, in those tweets so we removed them.

Stopwords: Stop words are commonly used words, in the English language, (such as: the, a, an, as, for, etc.) that can be ignored as they don't add significant meaning in sentences. We removed these words, using `nltk.corpus` which consists of corpora of English stopwords.

Urls Emojis and spaces: We also removed, urls or links to websites, which are not any contributing factor to the analysis, also we needed to remove the emojis blank spaces.

Removal of Non-English Tweets: We removed all the tweets that were not in English, after collecting them. As there is not so many good analysing tools available so we had to limit ourselves in working with English tweets only.

2.2.3 Tokenization

In this process, we split a sentence, phrase or an entire text document into smaller chunks such as, smaller terms or words. Let us consider an example. Given a string 'This is our house' when performed tokenization we get ['This', 'is', 'our', 'house'], analyzing these we can easily interpret the meaning of the text.

2.2.4 Stemming & Lemmatization

Stemming and Lemmatization helps simplifying the text data, by reducing the inflectional forms in the words, stemming is a crude way to simplify words to their root by cutting of some characters from the beginning or end, Lemmatization is a more systematic way of reducing words, compared to stemming, it considers the vocabulary and morphological structure of words. For example if we want to reduce the words "Studies" stemming would simply chop off the 'es' part at the end, and reduce the word into 'studi' or something similar, whereas in lemmatization it will be reduced to 'study' which is the correct meaningful form of the root verb.

2.3 Preprocessing Text Data: Feature Extraction

After cleaning the text data, we need to extract numerical features (Feature Vectors) to perform our further analysis. Any statistical or mathematical model does operate on string or text but numbers. So we need to convert the text to numbers in a way so that they can represent the relative importance of a word in a sentence to represent the sentiments expressed in it.

In this project we have applied 5 different text transforming techniques on our text data and have worked with all 5 dataset as we don't have a prior knowledge about which text transforming techniques would work best. The text-transforming techniques are described below:

2.3.1 Bag of Words Model

Bag of Words (BOW) model is one of the simplest vector space representational model for unstructured text-data. Vector space representation is a way of representing unstructured text data into numeric feature vectors, this model represents each text document as a numeric vector where each dimension represents a particular word from the corpus is present or not or the number of times it is present.

2.3.2 N-gram Model

An n-gram model, similar to BOW model, considers chunk of n words at a time and looks for it in the corpus and thus represents a text or document as a numeric vector.

2.3.3 TF-IDF Model

The BOW model simply counts the occurrences of words into a text or document, and does not consider which words are more important or less. TF-IDF model takes that into consideration.

Term Frequency: tf-idf is abbreviation for Term Frequency-Inverse Document Frequency. Term frequency measures frequency of word in a document. tf(t,d) i.e. relative frequency of term t within document d is given by:

$$\begin{aligned} tf(t,d) &= \frac{\text{Count of a word in document}}{\text{Number of words in the document}} \\ &= \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \end{aligned}$$

where $f_{t,d}$ is the raw count of a term t in a document d, i.e., the number of times that term t occurs in document d.

Inverse Document frequency: Inverse Document Frequency is obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient. $idf(t,D)$ is the ratio defined as,

$$idf(t,D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

where,

- N: total number of documents in the corpus.
- $|\{d \in D : t \in d\}|$: number of documents where the term t appears.If the term is not in the corpus, this will lead to a division-by-zero. It is therefore common to adjust the denominator to $1 + |\{d \in D : t \in d\}|$.

Term frequency–Inverse document frequency: Taking multiplicative value of TF and IDF we get TF-IDF score. tf–idf is calculated as

$$tf - idf(t,d) = tf(t,d) \times idf(t,d)$$

2.3.4 TF-IDF N-gram Model

This is an extension of N-gram model where we generate features by taking n contiguous words together from a sentence and calculate the value for each feature using TF-IDF score.

In this case we have taken the value of n=2.

2.3.5 TF-IDF Mixed N-gram Model

This is same as the previous TF-IDF N-gram model but here we take into account all the N-gram for n ranging to 1 to n. That is here our feature is all the unigrams(n=1), bi-grams(n=2),...,n-grams and again we calculate the score for a particular feature using TF-IDF values.

In this case we have taken $n = 3$, i.e. here the features are all the uni-grams,bi-grams and tri-grams together.

3 Dimensionality Reduction

If we have a large number of variables in our data set, that can pose the threat of model overfitting. Also, it becomes too difficult to cluster high dimensional data and model with too many features is hard to interpret. This phenomenon is called **Curse of Dimensionality**. In that case, we want to reduce the dimensionality of our data sets so as to work with fewer variables. There are primarily two ways of dimensionality reductions:

- Feature Elimination
- Feature Extraction

In Feature Elimination we drop one or more variables from our data set and this leads to loss of information from our original data. **Principal Component Analysis** is a popular dimensionality reduction technique where we try to formulate new variables taking the linear combinations of the original features in such a way that maximum variability is retained. This is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still retains most of the information in the original data set. It can also be used for visualization purposes for relatively lower dimensional data.

3.1 Principal Component Analysis

1. Let X be our design matrix of order $n \times p$, where n is the total number of observations and p is the number of features. For the first step of applying PCA, we need to standardize each of the feature vectors in order to make sure that any one of them with larger values (and so with larger variance) gets more importance than the others. Let Z be the matrix we get after standardization.

2. We consider $S_{n-1} = \frac{1}{n-1}Z'Z$ which is a symmetric and positive definite matrix. S_{n-1} is the sample variance-covariance matrix of our data set. Let the spectral decomposition of S_{n-1} be PDP^{-1}, where P consists of orthogonal eigen vectors of S_{n-1} and D is a diagonal matrix with diagonal elements as the eigen values.

3. Let, the eigen values $\lambda_1, ..., \lambda_p$ are sorted in descending order i.e. $\lambda_1 \geq \lambda_2 \geq .... \geq \lambda_p$ and similarly the eigen vectors are also re-arranged and we get P^*.

4. Now we calculate $Z^* = ZP^*$ and Z^* becomes our new data matrix with p number of new features. Also, the new features are uncorrelated linear combination of the old ones i.e

$$corr(Z_i^*, Z_j^*) = 0, i \neq j, i, j = 1(1)p.$$

   where, Z_i^* is the i^{th} column of Z^*. The first variable Z_1^* is called the first principal component explaining maximum variability, the second variable Z_2^* is called the second principal component explaining second most variability and so on as the $var(Z_i^*) = \lambda_i, i = 1(1)p$.

5. For selecting the number of principal components(K) to choose, first of all, we specify a threshold value $\alpha, 0 \leq \alpha \leq 1$ which is basically the proportion of variability we want to retain. The proportion of variability explained by first k principal components can be calculated by

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{p} \lambda_i}$$

Now, we choose minimum of such k's, such that proportion of variability explained by first k principal components is greater than or equal to α i.e

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{p} \lambda_i} \geq \alpha$$

6. Let k^* be the number of principal components we decided to keep. Then we take only the first k^* columns of Z^* and proceed further.

3.2 Application on our dataset

We standardize each of the feature vectors column-wise, for the 5 methods used before. We then applied PCA on each of the data-matrix retaining 90% of the original variation. After applying PCA we have been able to reduce the feature space significantly without losing much information. Before applying PCA the number of columns of our data matrix were 1233, 381, 226, 381 and 814. After applying PCA we ended up with 648, 229, 174, 249 and 523 columns respectively.

4 Cluster Analysis

Clustering is an unsupervised process used in multivariate data mining technique to group objects based on their similarities. A good clustering method will provide high intra-class similarity and low inter-class similarity. Quality of clustering is also measured by its ability to discover hidden patterns in the data. Some of its application are in biology and marketing field where we group flora and fauna given their features or customer segmentation based on their data of past buying records. There are several similarity and dissimilarity metric to measure "goodness" of a cluster such as Minkowski measure of distance, Cosine similarity, Mahalanobis Distance. Some of the major clustering approaches are:

- Partitioning algorithms: Here we construct a partition of a database D of m objects into a set of K clusters such that it optimizes the chosen partition criterion. K-Means algorithm is an example of this type of clustering.
- Hierarchy algorithms: we make a hierarchical decomposition of the dataset according to some criterion. In this algorithm, we develop the hierarchy of clusters in the form of a tree-like structure known as the dendrogram. This clustering technique has two approaches: Agglomerative or bottom-up approach and Decisive or top-down approach.
- Density-based: It is a clustering approach Based on connectivity and density function. Here we locate region of high density seperated by region of low density. DBSCAN clustering method is an example of this.
- Grid-based: This method works with a multi-resolution grid data structure on which clustering operations are implemented.
- Model-based: It is a statistical approach where the observed multivariate data is considered to have been created from a finite combination of component models. Each component model is a probability distribution, more generally a parametric multivariate distribution.

For our analysis purpose we have taken *K-Means Clustering Algorithm* approach to find the hidden patterns in our dataset. The algorithm is described as follows:

4.1 K-Means Clustering Algorithm

Let us consider the objects X = $\{x_1, x_2, ..., x_m\}$ and k no of output clusters S = $\{S_1, S_2, ..., S_K\}$. Our goal is to minimize WCSS (within cluster sum of square) given by

$$\arg\min_S \sum_{i=1}^{K} \sum_{x \in S_i} ||x - \mu_i||^2$$

$$\text{i.e.} \arg\min_S \sum_{i=1}^{K} \sum_{x \in S_i} d(x, \mu_i)$$

where μ_i is the mean of the datapoints in cluster S_i and $d(x, \mu_i)$ is the distance metric of the x's belonging to the class S_i from their mean μ_i. So, for a given K

- **Step 1:** First we randomly choose K datapoints as centroids(cluster centre) initially.
- **Step 2:** We calculate the distance of each datapoint from the centroids and assign the datapoint to its closet centroid.
- **Step 3:** Take average of the datapoints in newly formed clusters and reassign them to be our new centroids.
- **Step 4:** If the convergence criterion does not meet we repeat step 2 and step 3.

So, we continue the process until

- The newly assigned centroid remains the same.
- The datapoints in a cluster do not change after re-allocation.
- Maximum number of iterations is reached.

Now to proceed with this clustering technique we have to first choose K i.e. optimal number of clusters.

4.2 Choosing Optimum Number of Clusters

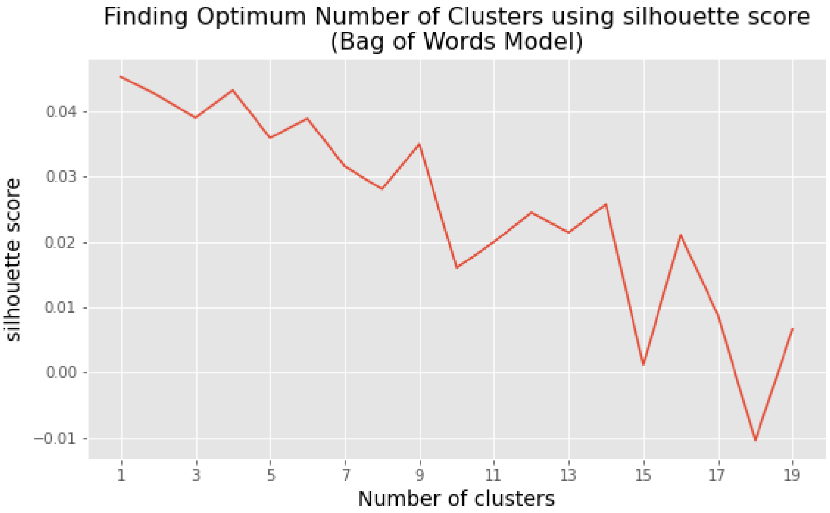
For selecting the optimum number of clusters we follow two approaches as described below:

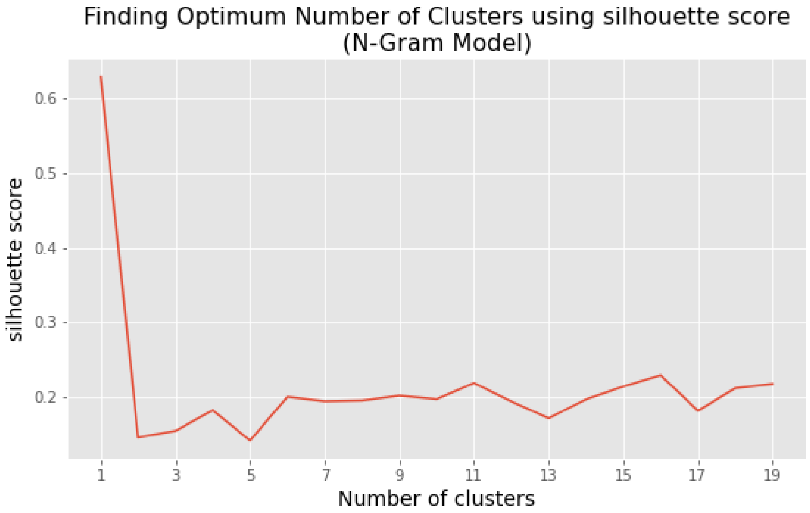
4.2.1 Silhouette Method

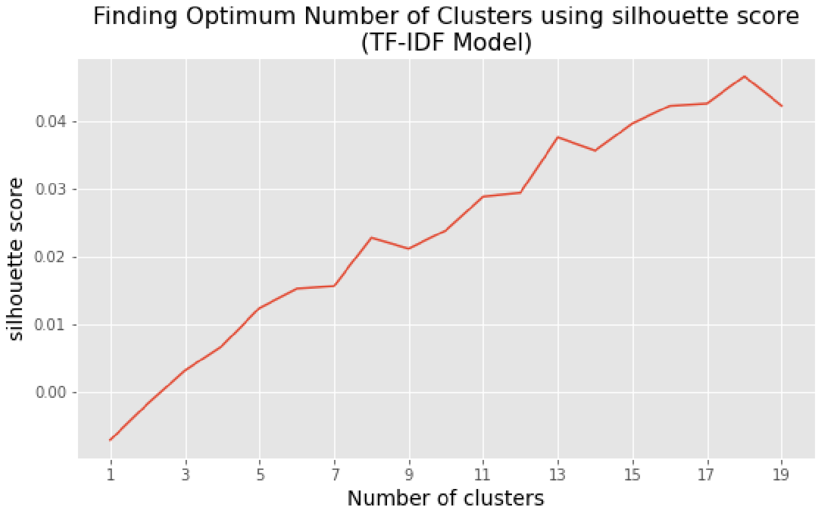
The silhouette coefficient can be thought of a measure of how similar a data point is within-cluster compared to other clusters. The equation for calculating the Silhouette coefficient for a particular datapoint is given by, $S(i) = \frac{b(i)-a(i)}{max\{a(i),b(i)\}}$ where,

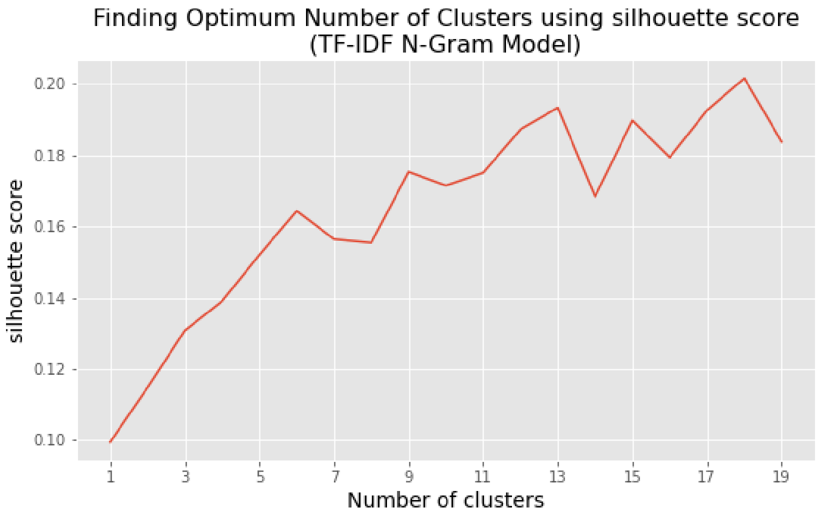
- S(i) is the silhouette coefficient of the data point i.
- a(i) is the average distance between i and all the other data points in the cluster to which i belongs.
- b(i) is the average distance from i to all clusters to which i does not belong.

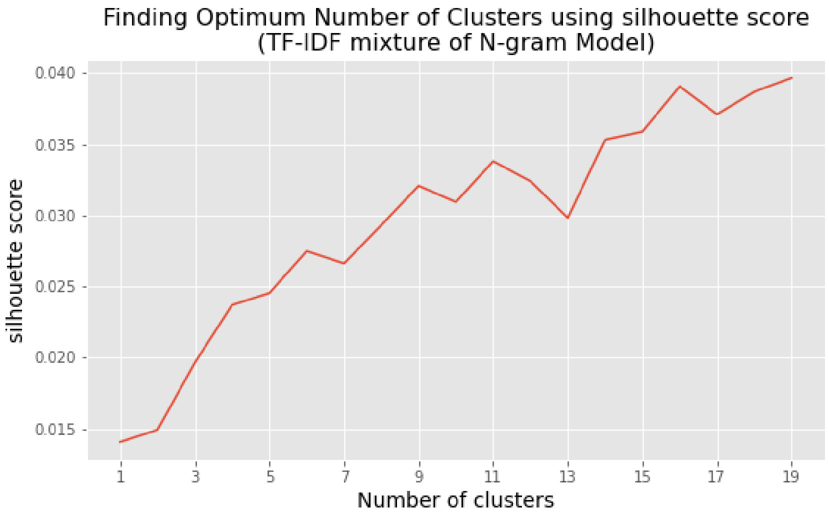
Next we calculate average Silhouette score considering all the datapoints and plot it against the number of clusters. The value of the silhouette coefficient lies between [-1, 1]. Also, when the value is one for a single datapoint, it means that the datapoints within the cluster the i^{th} point is assigned to, is very compact. The worst value $S(i)$ can take is -1. We choose that k for which the average Silhouette score over all the datapoint is maximum.











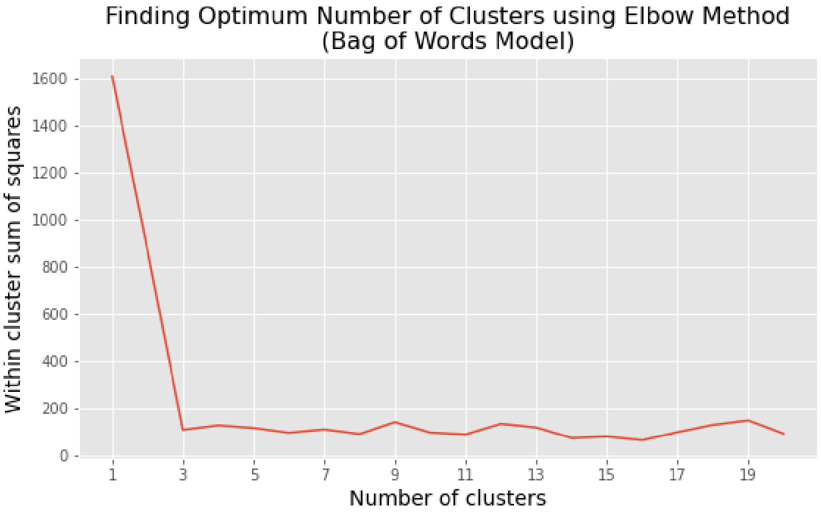
In general we choose that value of k as the optimum number for which the average silhouette score over all the samples is maximum. But in this case the graphs from all the 5 dataset didn't give us a clear picture of the optimum value for k. So we moved to our next method for choosing the optimum value for k.

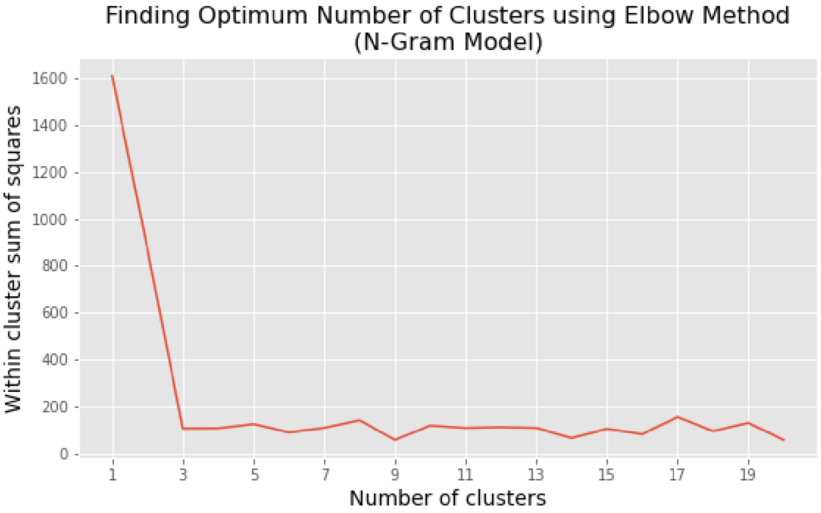
4.2.2 Elbow Method

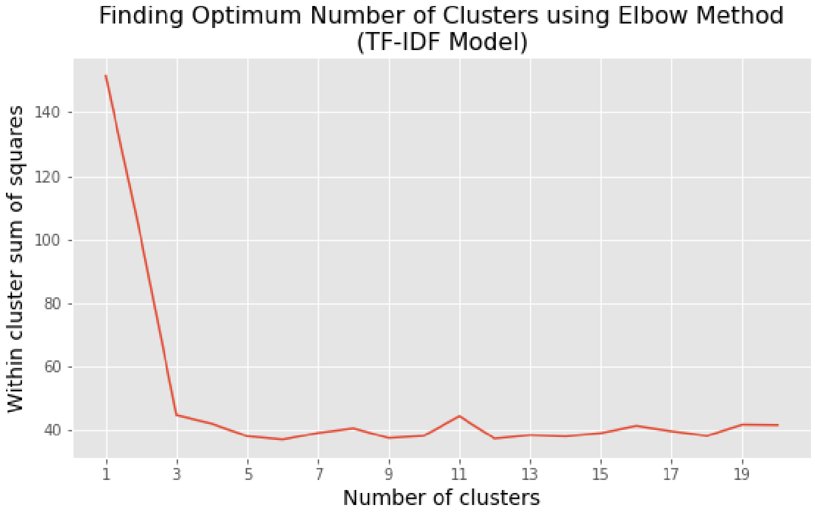
Elbow Method is one of the oldest method to find the optimal no. of cluster in K-Means Clustering Algorithm.In this method,

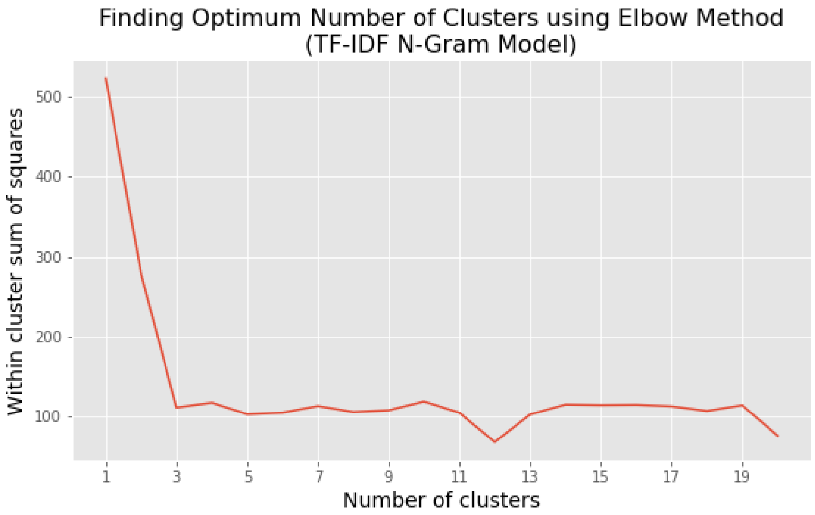
1) For a range of k,we fit K-Means Clustering Algorithm.
2) find Within Cluster Sum of Squares(WCSS) by equation(1) for each k.
3) Then, we plot a lineplot taking **number of clusters(k)** in horizontal axis & **WCSS** in vertical axis.

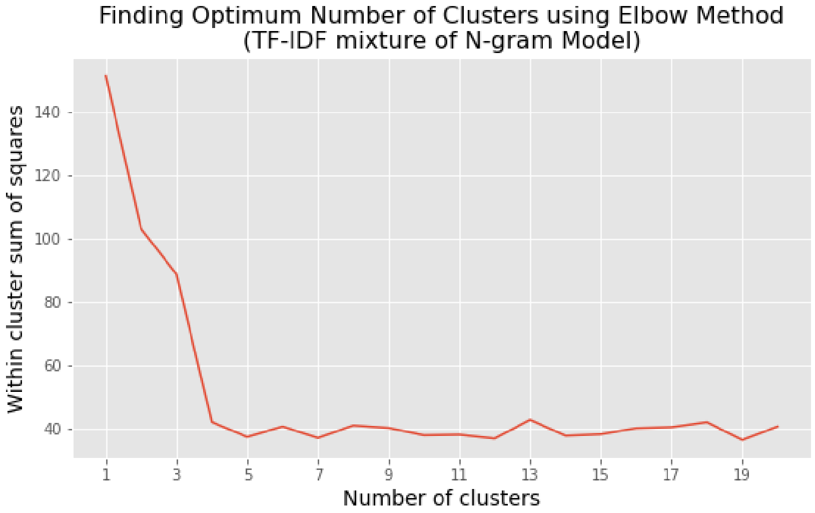
Now, the Elbow point of the plot is that point after which decrease in WCSS is not worth of increasing number of clusters. We choose the elbow point as the optimal number of clusters.


Finding Optimum Number of Clusters using Elbow Method
(Bag of Words Model)
Within cluster sum of squares
0
200
400
600
800
1000
1200
1400
1600
1
3
5
7
9
11
13
15
17
19
Number of clusters


Finding Optimum Number of Clusters using Elbow Method
(N-Gram Model)
Within cluster sum of squares
0
200
400
600
800
1000
1200
1400
1600
1
3
5
7
9
11
13
15
17
19
Number of clusters


Finding Optimum Number of Clusters using Elbow Method
(TF-IDF Model)
Within cluster sum of squares
40
60
80
100
120
140
1
3
5
7
9
11
13
15
17
19
Number of clusters





From these graphs above we can clearly see an elbow is formed at k=3 for almost all of the dataset. After the value 3, the reduction in **Within Cluster Sum of Squares** is very negligible. So we choose 3 as the optimum number of clusters in this problem.

5 ANALYSIS AND RESULTS

After fitting a K-Means clustering model with optimum number of clusters equals to 3, we assigned each tweets to one of these 3 clusters and got the class label for each tweet. Then observing a subset of tweets in each cluster we can make a guess about the types of tweets that are coming from each cluster.

- **Tweets from the first cluster:**
  1. “I’m absolutely horribly stressed with all of the news of Ukraine and Russia.”
  2. “Russia keeps bombing residential / civilian structures in Ukraine.”
  3. “Russia Ukraine war: Will Russia use nuclear weapons?”
  4. “Joining Apple and others, Microsoft stops sales in Russia amid invasion: ‘We stand with Ukraine’ ”

5. "While Russia was removed from the swift system, many Russian banks faced sanctions."
6. "Bitcoin price: Why is crypto down today? Crypto news and prices of BTC, Ethereum, Solana as Russia invades Ukraine."

By observing a subset of tweets from the first cluster we can see that these tweets talk about the Russian invasion and it's economic effect. These tweets represents fear and some of the tweets also condemned this act.

- **Tweets from the second cluster:**

  1. "Controversial take: Ukraine needs to surrender now to prevent further loss of life, they simply cannot win and the west will not directly enter military conflict with Russia. I'm not saying this is good, but what must happen to prevent more innocent Ukrainian deaths."
  2. "Let's remember that all of this was avoidable. Russia negotiated in good faith for 8 years, made its concerns known, and pushed for a peaceful resolution. Both Ukraine and the West were categorically uninterested in resolving the conflict in Donbas."
  3. "The population of the United States is growing and we the people of the United States need more fertile land. Now that the opportunity is knocking at our door. Let's take Russia and all its land!"
  4. "I hope the war between Russia and Ukraine will end soon But why has the world forgotten Yemen????!!! Today is 253th day of the war in Yemen."
  5. "@PMOIndia @narendramodi Our govt. Stand on this war is very clear and logical. Those who don't understand, just think who is Ukraine? and Russia. What was their stands towards india. Answer is simple. Russia and prez. putin Rocks "Akhand bharat" ka sapna ab dur nahi."

  Tweets in this segment are little bit encouraging and indirectly supporting Russia's aggression. People are somehow justifying this aggression by mentioning their own references. However ratio of tweets coming from this cluster is lowest amongst the three.

- **Tweets from the third cluster:**

  1. "In fact this war is being fought between Russia and USA, They both choose Ukraine as a battleground."
  2. "It's because they are not like USA who would have killed already hundreds of thousands at this stage Russia is considering a big part of Ukraine as ethnic brothers. The goal is to destroy the infrastructure of NATO."
  3. "US President Biden Joins Emergency NATO Session on Russia's Ukraine Invasion President Joe Biden joined an emergency NATO summit Friday to strengthen the frantic Western response."
  4. "Live: President Biden is taking pre-approved questions from the press about Russia's invasion of Ukraine."
  5. " @dhruv_rathee So ladies and gentlemen welcome to the world of pseudo activism, Not a single tweet or video on how Indians and Africans are being mistreated and did

back-to-back videos on how Russia invaded Ukraine. Not a single video on US bombing Somalia oh I forgot they are not whites"

6. "@BorisJohnson All you're offering Ukraine are words &amp; sanctions that will have no impact whatsoever on Putin's actions. You seem to view Russia's invasion of Ukraine as a "quarrel in a far away country, between people of whom we know nothing".
7. "520,000+ refugees have fled Ukraine since Russia waged war."

From the third cluster we can observe tweets not particularly in support of Russia or Ukraine but most of these tweets are targeting USA and some of it are generally conveying some news regarding this matter. Ratio of tweets coming from this cluster is highest amongst the three.

6 CONCLUSION

After performing a detailed analysis of the tweets regarding Russia-Ukraine war in the timeline from 24^{th} February to 9^{th} March 2022, we got few insights about the sentiments expressed by these tweets. Not all of the tweets falling in the same cluster is expressing the absolute same thought but we can get a overall idea of the sentiments in a cluster of tweets which are mainly: **'Condemning and discussing the adverse effect of the war'**, **'Someway or other justifying the invasion'**, and *'other tweets'* which mainly consists tweets talking about other countries other than Russia and Ukraine or just the news regarding the war and not particularly biased to one of the country.

7 FURTHER SCOPE OF WORK

In this project we have faced some difficulties while collecting and transforming this huge scale data. There are certain areas that can be improved if our time and resources were not a constraint.

- First of all we have collected 2.8 million tweets to analyse. It took a lot of time to process and build a model on top of this dataset. So we took a random sub-sample from this data to do our further analysis which is half as good. It may happen that we could get more number of patterns in our dataset if we have worked on the whole dataset. We couldn't do so because of the lack of resources and memory constrain at our workspace.
- During the text cleaning procedure we removed all the tweets containing non-English words and lost almost 2/3rd of our data in this process. If we knew a better way to translate non-English tweets to English to maintain uniformity then we could work with more number of data and our work would be more prominent in that case.

8 SUPPLEMENTARY MATERIALS

For more details regarding the codes and data analysis the readers are directed to the GitHub repository. We have also taken help of some of the articles listed below:

- Cluster Analysis
- Bag of Words and Tf-Idf
- Sentiment Analysis