Sentiment Analysis of Russia Ukraine Invasion

Ajit Jadhav

Srashti Soni

Karan Ajay Pisay

Shree Sharma

University of Maryland, Baltimore County

Data 602 : INTRODUCTION TO DATA ANALYSIS AND MACHINE LEARNING

DR. TONY DIANA

05/13/2022

Abstract

The purpose of this paper is to get a sentimental understanding on the recent invasion of Ukraine by Russia. Through this paper we aim to analyze the emotions of the world news and the tweets that were posted on twitter on keywords related to the invasion. Such as ‘Russian invasion of Ukraine ', ‘Ukraine war’, ‘Russia war’, ‘Ukraine vs Russia', ‘Russia-Ukraine war’, ‘Russia-Ukraine conflict’. We investigated the subject of sentiment analysis of the Ukraine and Russia conflict. In this research, created a corpus of Ukrainian and Russian news, and labeled each text with one of three categories: positive, negative, or neutral. The hypothesis of the overall project is that we expect the overall sentiment to be negative.

Introduction

The Russo-Ukrainian War is a conflict between Russia and Ukraine that began in February 2014. Following a Russian military build-up on the Russia–Ukraine border beginning in late 2021, the conflict escalated substantially on February 24, 2022, when Russia launched a full-scale invasion of Ukraine (Ames & Kononenko, 1959).

Our Sentiment analysis project involves two datasets, one being the Google news and other being the tweets on Twitter. The reason behind using two datasets is to gauge the overall sentiments across the major platforms one being news and Twitter. The stages in our projects are, gathering Twitter data using the Twitter API and fetching the Google news for the past six months from the month of November, 2021 till May, 2022, Preprocessing the data, and classifying the text as positive, negative, and neutral, and building a classification model. Based on our initial research, Naive Bayes Classifier is a popular supervised machine learning classification model, it uses the conditional probability to classify the words into their respective categories that is positive, negative, or neutral.

For further analysis and testing our datasets on multiple classification models we constructed a pipeline of machine learning models, such as, Decision Tree Classifier, Logistic Regression, Random Forest Classifier, Ridge Classifier, SVC, and AdaBoost classifier models. The idea behind running different models using the pipeline is to build a model which gives us the highest accuracy in predicting the sentiment category of the text.

Our Hypothesis

- The hypothesis of the overall project is that we expect the sentiment around the globe to be towards negative.
- Another Hypothesis would be, Twitter being a social media platform, we expect the tweets on Twitter to be on the negative side.
- What are the effects of invasion of Ukraine by Russia on the sentiments of people around the world ?
- What are the sentiments before the invasion and after the invsion happened ?

Methodologies

As mentioned earlier in the introduction we have used two datasets that is the Twitter data and the Google news data. As the topic is sentiment analysis and text classification the "tweets" column was the column of focus for Twitter data and "title" column for Google news data.

The Twitter data has the following columns:

1.Userid
2.Username
3.Acctdesc
4.Location
5.Following
6.Follower
7.Totaltweets
8.Usercreatedts
9.Tweetid
10.Tweetcreatedts
11.Retweetcount
12.Tweet

13.Hashtags

14.Language
15.Coordinates
16.Favorite_countextractedts

The Google news data has been divided into three datasets:

- Before invasion
- After invasion
- Combined data of four countries (UK, USA, INDIA, CANADA)

The columns for Google news data are as follows:

1. Title
2. Link
3. published_dates

This project entails sentiment analysis on the Russia-Ukraine war using the tweet and the google news collected over a period of a certain timeframe. The Sentiment analysis is implemented using python and the overall chronology of the project is as follows:

- Exploratory Data Analysis
    - An exploratory data analysis will be carried out to ensure that the influence of war on people's thinking was primarily negative. Confirming people's opinions and the overall country-by-country distribution of emotions will provide us with valuable insight into what was going on in people's minds before and after the conflict.
- Sentiment Analysis:
    - After all of the data has been analyzed, we will do sentiment analysis on data from Twitter and Google News to determine the wave of emotion before and during the war. Multiple datasets will be subjected to sentiment analysis. Because the datasets are diverse, the research will be more accurate overall.
- Machine Learning:

- On both datasets, we have used machine learning algorithms and made use of wonderful models and libraries like BERT, Pipelines, Decision Tree Classifier, Logistic Regression, Random Forest Classifier, Ridge Classifier, SVC, and AdaBoost classifier models.

Literature Survey

The aim of our literature review was to gain insights into Twitter and Google News as platforms in the context of sentiment analysis. We have referenced two research papers that are continuously tracking public discourse on social media about the Russia-Ukraine war.

In a paper on sentiment analysis of Russian-language content, the author focuses on the Ukraine-Russia conflict and immigration issues (Smetanin, S, 2020). The author has analyzed the text to gauge the sentiment of the content. Previously, Using the "StreamKM++" method, Twitter sentiment analysis was used to anticipate the Maidan upheaval in Ukraine (Iana Sabatovych, 2019). The model had an accuracy of 96.75% in predicting social movement.

We referenced another paper namely “Sentiment Analysis in the Ukrainian and Russian News" by Victoria Bobichev published in November 2017. Their data for analysis contain news data from two countries. The paper entails sentiment analysis on the conflicts that were held between Russia and Ukraine. Experiments conducted in their research contain classification of the text with 4 diverse models namely Naïve Bayes, DMNBtext, NB Multinomial and Support Vector Machine. Furthermore, for news in Ukrainian and Russian languages, their testing results suggested an average F1-score of 0.82. (link : https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8100410)

With respect to our research, our data contains news from across the globe and tweets from Twitter. Moreover, our research also focused on multiple timelines that is before the invasion and after the invasion to analyze the sentiments of the duration. Apart from the news and tweets, we gathered and merged the data on this invasion from 4 countries like USA, UK, Canada and India to get the sentiments of these countries on the invasion as Russia has strong business relations with these countries.

For analyzing the twitter data we have used Naïve Bayes as the base model and implemented Decision Tree Classifier, Random Forest, Ridge Classifier, adaboostclassifier, Logistic Regression and SVC different models by constructing a pipeline. Using this technique, we were able to produce an accuracy of 89% through Ridge Classifier.

Moreover, for analysis of google news, we have successfully implemented the pipeline for model draining through Decision Tree Classifier, Random Forest, Ridge Classifier, adaboostclassifier, Logistic Regression and SVC. Amongst the mentioned models, Decision Tree Classifier outperformed the other models with an accuracy of higher than 85%. To further improve our model, we have used transformers to train the model using BERT methodology which resulted in optimal accuracy of 93%. As seen in the cited research papers, the researchers have implemented the sentiment analysis using classification models, whereas we have implemented sentiment analysis models through a pipeline of various classification models and to further improve the accuracy we have used transformers as well.

Pre-Processing the Data

The preprocessing has been further divided into three sections i.e., Data Gathering, Data Scraping, and Data Cleaning and is explained below.

Data Gathering

For Twitter Data: Data Gathering is one of the most crucial parts of analysis and to begin with our research and analysis we followed the ideology of analyzing the tasks from two different points of view. One is the sentiment of the tweets that are posted on Twitter which is a social networking and news website where users exchange brief messages known as tweets. Tweeting is the practice of sending brief messages to everyone who follows you on Twitter in the hopes that someone would find your remarks useful and engaging. Microblogging is another way to describe Twitter and tweeting. To keep things scannable, Twitter has a conscious message size restriction: each microblog tweet entry is limited to 280 characters or fewer. This size restriction encourages targeted and sophisticated language use, making tweets easy to peruse yet difficult to produce. Twitter became popular as a result of this size constraint. Taking this fact to our advantage we based our research on fetching the tweets using the Twitter API and analyzing the data to understand more about the sentiments of the people who tweet on Twitter regarding the Russo-Ukrainian War (*What Is Twitter & How Does It Work?*, 2021).

For Google News Data: To get a better insight on what is the sentiment of the whole world, we began with analyzing the news articles from google news over a time period of the past six months to get an absolute idea of the sentiments and understand the emotion of people due to this atrocity. With Google News, we were able to discover current events, worldwide news, and diverse content from different publishers. Google News is a news aggregator service developed by Google. For this sentiment analysis we gathered the data for the past six months i.e., November 5th, 2021 till May 10th, 2022 by using the "PyGoogleNews" in which with the help of keywords we were able to gather the news from various publishers around the globe. We have also fetched the data for a before the war and after the war analysis to conclude what were the sentiments before the invasion that occurred on February 24 and what were the sentiments after it and how it changed the emotions of the people in the whole world (*What Is Twitter & How Does It Work?* 2021).

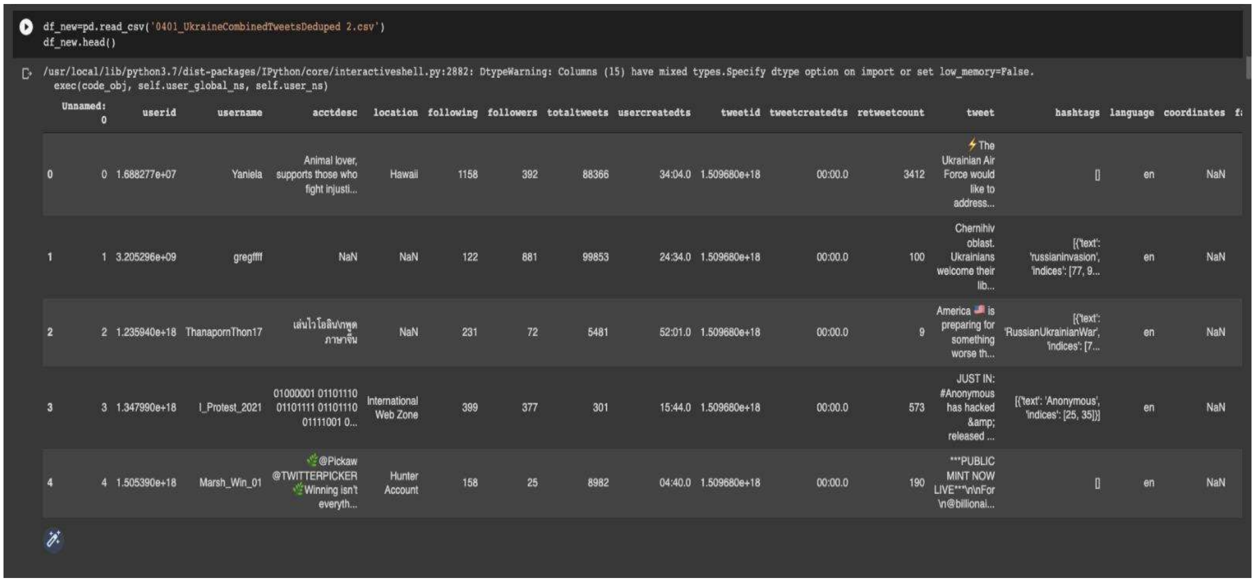
Data Scraping:

For Twitter Data: To scrape the data from Twitter we have used the Twitter API, we have employed the Tweepy library available in python to import this dataset. Tweepy is an open-source Python program that makes it extremely easy to use Python to access the Twitter API. Tweepy is a library of classes and methods that reflect Twitter's models and API endpoints, and it transparently handles implementation details such as Data encoding and Decoding, HTTP requests, OAuth authentication, rate limits, and streams. According to our research, gathering and scraping the data using any other methodology would have made us care about the low-level details such as HTTP requests, data serialization authentication and rate limits which may be time-consuming and error-prone. Instead, Tweepy allowed us to concentrate on the features we wanted to create (Real Python, 2021).

Tweepy allows users to extract custom data about tweets with users having the flexibility to filter data based on keywords, hashtags, usernames, search terms, languages, geographic area, date intervals, etc. Our research has focused on keywords such as 'Ukraine war', 'Russia war', 'Ukraine vs Russia', 'Russia-Ukraine war', 'Russia-Ukraine conflict' 'Russian invasion of

Ukraine. We downloaded the data and uploaded it on GitHub. The data was extracted for several days in order to maintain track of it and guarantee that it remained current.

For scrapping the Twitter data a few credentials are required such as “API key and Secret".which is the username and password for the app that we have created on the developer platform. Moreover, we require a user access token which represents the user on whose behalf we are making the request and similarly, an apt token is required for making a request to an endpoint that requires OAuth 2.0. Nonetheless, there are different access levels and versions involved in using the Twitter API, we have the ‘Essential’ access which is free of cost and allows us one project that lets us access up to 500K tweets per month.



```
df_new=pd.read_csv('0401_UkraineCombinedTweetsDeduped 2.csv')
df_new.head()
```

```
/usr/local/lib/python3.7/dist-packages/IPython/core/interactiveshell.py:2882: DtypeWarning: Columns (15) have mixed types.Specify dtype option on import or set low_memory=False.
  exec(code_obj, self.user_global_ns, self.user_ns)
```

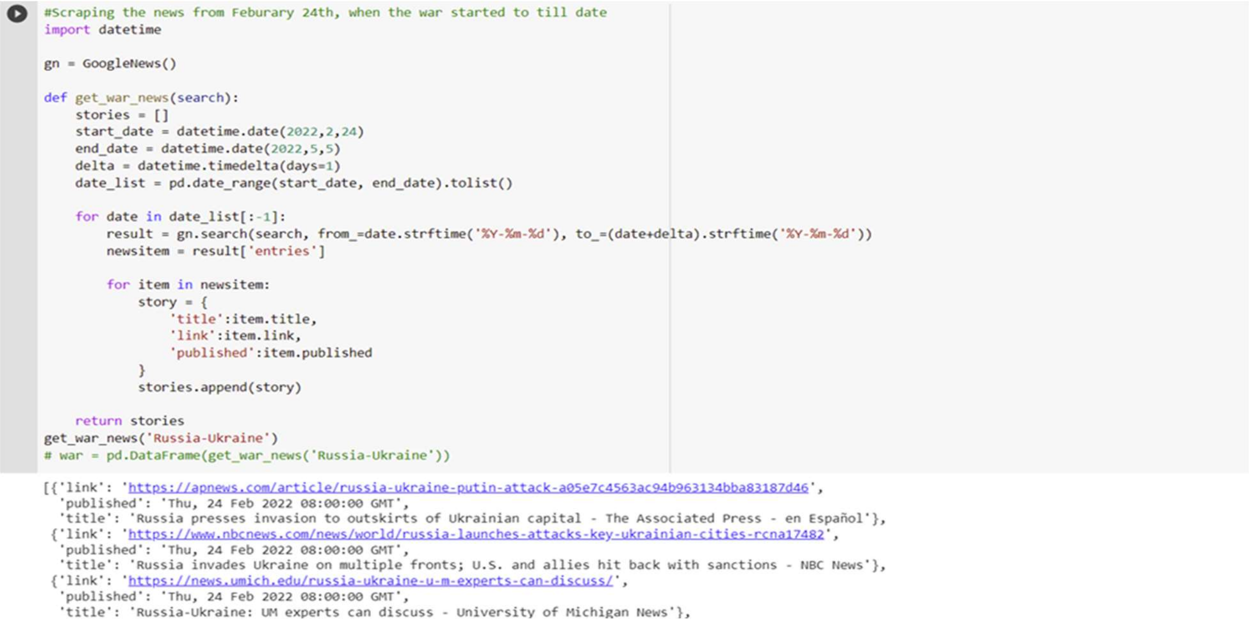
	Unnamed: 0	userid	username	acctdesc	location	following	followers	totaltweets	usercreatedts	tweetid	tweetcreatedts	retweetcount	tweet	hashtags	language	coordinates	f:
0	0	1.688277e+07	Yaniela	Animal lover, supports those who fight injusti...	Hawaii	1158	392	88366	34:04.0	1.509680e+18	00:00.0	3412	⚡The Ukrainian Air Force would like to address...	[]	en	NaN	
1	1	3.205296e+09	gregffff	NaN	NaN	122	881	99853	24:34.0	1.509680e+18	00:00.0	100	Chernihiv oblast. Ukrainians welcome their lib...	[{'text': 'russianinvasion', 'indices': [77, 9...	en	NaN	
2	2	1.235940e+18	ThanapornThon17	เล่นไวโอลิน\nพูดภาษาจีน	NaN	231	72	5481	52:01.0	1.509680e+18	00:00.0	9	America 🇺🇸 is preparing for something worse th...	[{'text': 'RussianUkrainianWar', 'indices': [7...	en	NaN	
3	3	1.347990e+18	I_Protest_2021	01000001 01101110 01101111 01101110 01111001 0...	International Web Zone	399	377	301	15:44.0	1.509680e+18	00:00.0	573	JUST IN: #Anonymous has hacked &amp; released ...	[{'text': 'Anonymous', 'indices': [25, 35]}]	en	NaN	
4	4	1.505390e+18	Marsh_Win_01	🌿@Pickaw @TWITTERPICKER 🌿Winning isn't everyth...	Hunter Account	158	25	8982	04:40.0	1.509680e+18	00:00.0	190	***PUBLIC MINT NOW LIVE***\n\nFor \n@billionai...	[]	en	NaN	

Fig: An overview of the Twitter Data Set.

For Google News Data: The PyGoogleNews library which was created by Artem Bugara allowed us to gather the data from google news. It is a python wrapper of the Google news feed and provides top stories, topic-related news feeds, a geolocation news feed, and an extensive full-text search feed. With PyGoogleNews we can try any combination of language and country that are supported by Google News. For example, if we want to search for the country “USA”, It will give us the choice of options for the language that is spoken in America and we can simply gather the data at your will using the in-built functions.

For the purpose of our analysis, we have gathered information on top news related to the Ukraine-Russia War starting from November, 22nd 20221 till February 22 2022. We have also fetched the news in regards to four countries to see the geolocational impact of the invasion and what are the sentiments of countries like the United States, Canada, United Kingdom and India which is reflected in their local news channels. However, to keep the study straightforward we have decided to go with English as the language parameter.

The PyGoogleNews is a dynamic, flexible and above all easy-to-use tool which allows us to search the news headline using the keyword parameter. The After war Data has been collected using the same methodology from February 24th till May 2022. Both the datasets use the "Russia-Ukraine" as a keyword in the keyword parameter. Below is a snippet of how the code works.



```
#Scraping the news from Feburary 24th, when the war started to till date
import datetime

gn = GoogleNews()

def get_war_news(search):
    stories = []
    start_date = datetime.date(2022,2,24)
    end_date = datetime.date(2022,5,5)
    delta = datetime.timedelta(days=1)
    date_list = pd.date_range(start_date, end_date).tolist()

    for date in date_list[:-1]:
        result = gn.search(search, from_=date.strftime('%Y-%m-%d'), to_=(date+delta).strftime('%Y-%m-%d'))
        newsitem = result['entries']

        for item in newsitem:
            story = {
                'title':item.title,
                'link':item.link,
                'published':item.published
            }
            stories.append(story)

    return stories
get_war_news('Russia-Ukraine')
# war = pd.DataFrame(get_war_news('Russia-Ukraine'))
```

```
[{'link': 'https://apnews.com/article/russia-ukraine-putin-attack-a05e7c4563ac94b963134bba83187d46',
  'published': 'Thu, 24 Feb 2022 08:00:00 GMT',
  'title': 'Russia presses invasion to outskirts of Ukrainian capital - The Associated Press - en Español'},
 {'link': 'https://www.nbcnews.com/news/world/russia-launches-attacks-key-ukrainian-cities-rcna17482',
  'published': 'Thu, 24 Feb 2022 08:00:00 GMT',
  'title': 'Russia invades Ukraine on multiple fronts; U.S. and allies hit back with sanctions - NBC News'},
 {'link': 'https://news.umich.edu/russia-ukraine-u-m-experts-can-discuss/',
  'published': 'Thu, 24 Feb 2022 08:00:00 GMT',
  'title': 'Russia-Ukraine: UM experts can discuss - University of Michigan News'},
```

Fig: Scraping the news from google news using pygoogle news.

The scraped result consists of the title, link, and publishing date of the article which is later stored in a Pandas Dataframe for further analysis and cleaning.

Data Cleaning:

For Twitter Data: After fetching the data from the Twitter API we found that there were a few impurities and noise in the acquired data which had to be removed for achieving higher precision inaccuracy. As a result, the Twitter data was preprocessed and cleaned in order to make it suitable for future studies. We have kept the main language as English to ensure the model is readable, definitive, and comprehensible. For this process, we made use of the ‘neattext’ library which was made available to us in python. NeatText is a straightforward Natural Language Processing tool for cleaning and preprocessing text data. It can sanitize phrases as well as extract emails, phone numbers, weblinks, and emoticons from them. It can also be used to create pipelines for text pre-processing. Similarly, we have made use of this library to remove hashtags, usernames, multiple spaces, punctuations, URLs and emojis present in the data.

For Google News Data: In the case of Google news data, we were aware of the fact that many of the articles are published by international channels and publishers hence, creating a possibility of duplication which would adversely affect our analysis. Therefore, we sanitized the data by removing the duplicated news article. We have utilized Snorkel to come up with heuristics and programmatic rules utilizing functions that assign labels to two classes that differentiate whether the headline is positive (1) or negative (2). Another set of labeling functions was created using the TextBlob tool, which is a sentiment analyzer that has been pre-trained. We have created a Pre-processor that runs TextBlob on our headlines before extracting the polarity and subjectivity ratings.

Initial Analysis

In the initial analysis, we have created a Word Cloud which displays the most occurring words in our datasets.

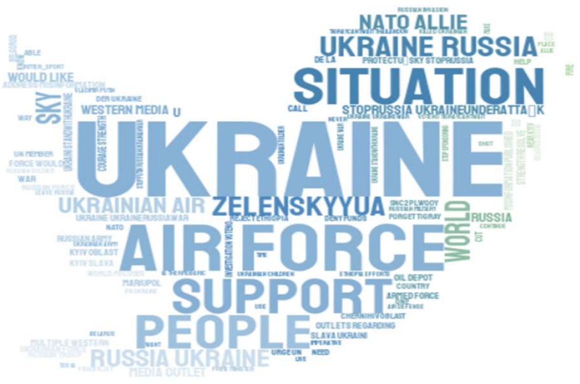


Fig: Word Cloud of the Twitter Russia-Ukraine war tweets.

Below is a bar graph representing the language distribution for the Twitter data where we can clearly see the dominance of the English language in tweets followed by french.

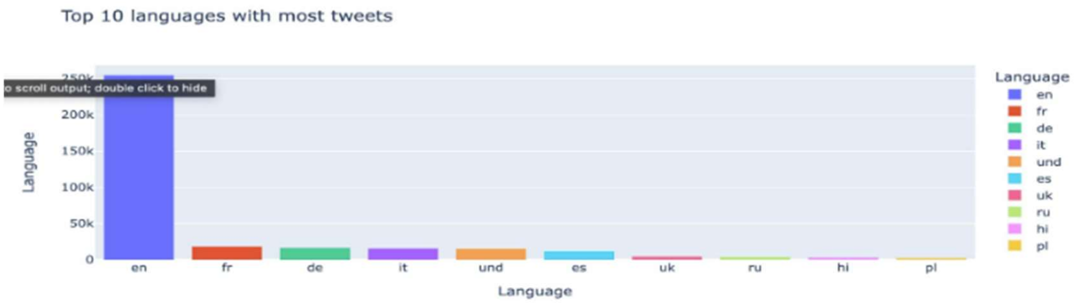


Fig: Maximum tweets were recorded in which language.

From the below plot we can see that naturally, Ukraine is at the top in terms of counts in tweets followed by the USA, India, and France.

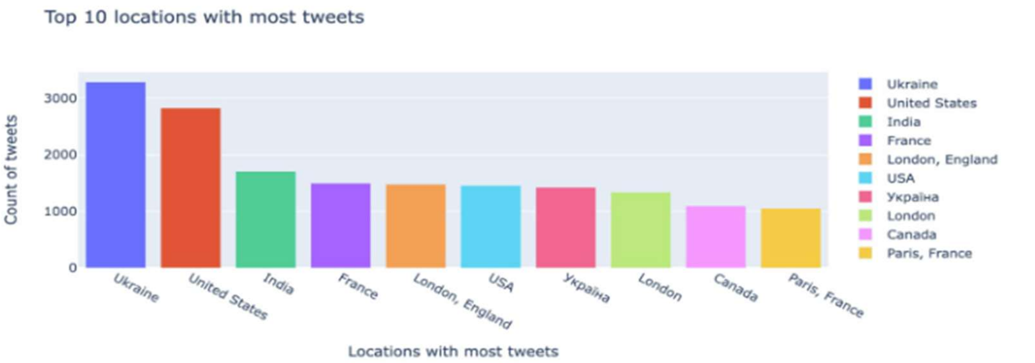


Fig: Top 10 Locations with the most tweets.

Sentiment Analysis on Twitter Data

Twitter sentiment analysis is an area that has recently piqued researchers' interest. Twitter is a famous microblogging site where people may share their thoughts and opinions. Sentiment analysis in Twitter addresses the issue of assessing tweets in terms of the opinions expressed in them. Sentiment analysis on Twitter data will assist in pointing out the emotiveness of these millions of tweets, as Twitter is a place where people express their opinions and thoughts about current events across the world. Sentiment analysis will not only reveal the emotional state of the persons behind the tweet, but it will also ensure that the decisions and outcomes are unbiased and free of human involvement. Sentiment analysis on Twitter data entails numerous steps:

Sentiment analysis comes next after confirming that the data has been cleaned and molded to meet the requirements. The positivity, negativity, and neutrality of the emotions behind the tweets are used to classify the data. Additionally, the Textblob library played an important role by providing API access to key functions like sentiment analysis and spelling correction.

To begin the sentiment analysis on the data, the data is tokenized, which means that the sentences extracted from the tweets are collected and then split into tokens, each of which consists of a single word. Textblob features some predetermined rules, or a word and weight dictionary, with some scores that help to assess the polarity of a statement. For a particular input text, the Textblob sentiment analyzer produces two properties: polarity and subjectivity. Personal opinion, emotion, or judgment are all examples of subjective sentences. We categorized the text as positive(1), negative(0), or neutral(-1) based on its subjectivity and polarity score using the snorkel library.

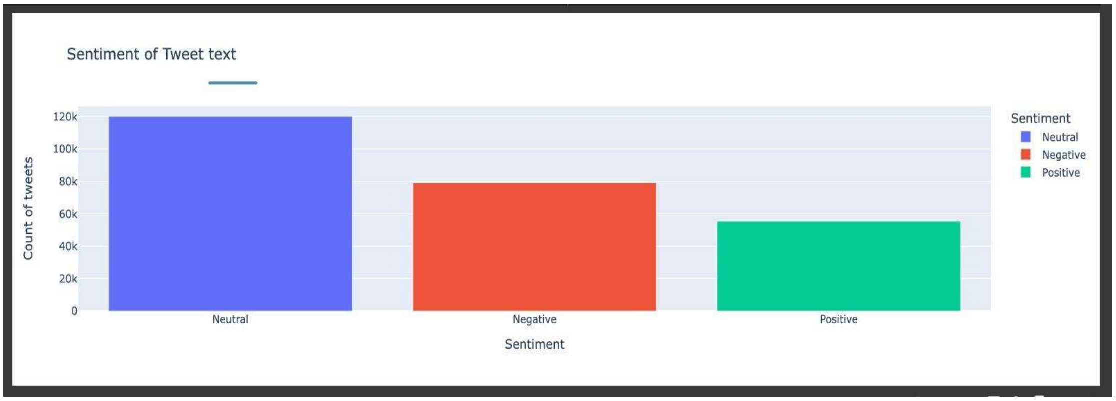


Fig: The sentiment of the Twitter Dataset.

We can observe from the following graph that the emotions reflected prior to the war were mostly neutral. The classification models were built to estimate the sentiment of tweet texts based on the data obtained from the visualization.

Sentiment Analysis on Google News

A great number of news pieces and headlines about current events in the Russia-Ukraine war are published on Google News every day. As we all know, this war has caused tremendous emotional and economic roller coasters, resulting in a slew of news stories and hence data for sentiment analysis.

We successfully scraped data from three sources and modified it to our specifications. We simply needed the text data for the sentiment analysis. As a result, we deployed the textblob library in Google News as well, taking into account all of the news titles. The snorkel library used three groups to categorize the data after determining the polarity and subjectivity of the data:

- (-1) is neutral
- (0) is negative
- (1) is positive

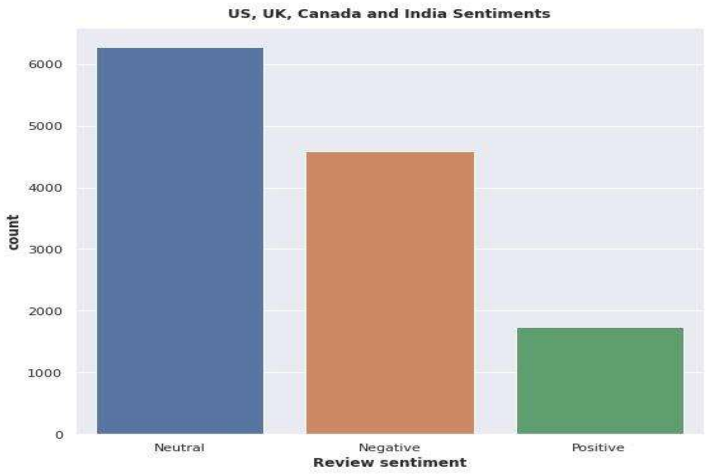


Fig: Sentiments through Google News Data (Country-wise)

As we can observe, taking the four countries into consideration, the overall emotion of the population was neutral. However, Positive emotion is far less compared to the other two.

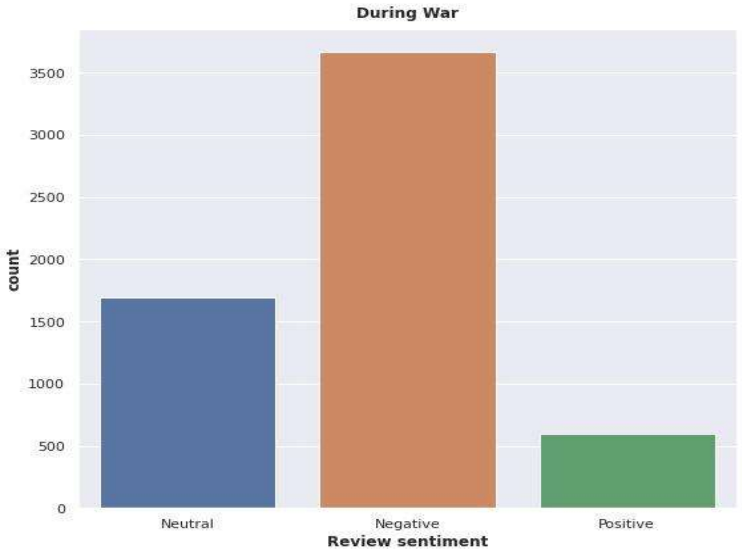


Fig: Sentiments through Google News Data (During War)

Machine Learning

Based on our initial research, we found that the Naive Bayes model is the best choice for text classification hence, The Bayes' Theorem is used to create a set of classification algorithms known as Naive Bayes classifiers. It is a family of algorithms that share a similar idea, namely that each pair of features being categorized is independent of the others. In Naive Bayes, the dataset is divided into two parts, namely, the feature matrix and the response vector. The feature matrix contains all the rows of the dataset and the response contains the value of the class variable of each row (Dr K & Dr.M, 2020).

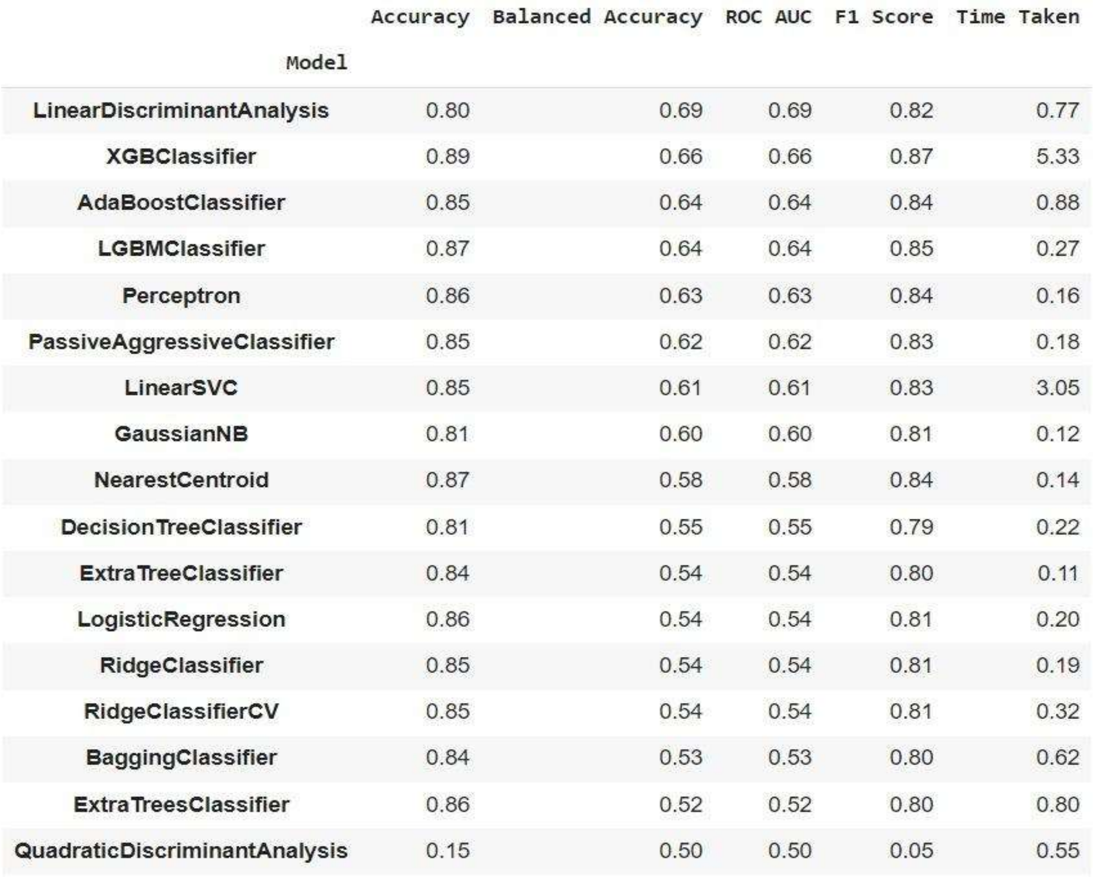
The main two assumptions of Naive Bayes are that the feature variable of the same class makes an independent and equal contribution to the outcome. The Bayes Theorem calculates the chance of an event occurring given the probability of a previous event. The mathematical proof of Bayes' theorem is as follows:

$$P(A \mid B)=\frac{P(B \mid A) P(A)}{P(B)}$$

Before implementing the pipeline, we executed the lazy predict method to know the accuracy of all the models related to our research and to find the top 6 models for our pipeline. In the pipeline we have implemented 6 different models namely:

1. Decision Tree Classifier
2. Random Forest
3. Ridge Classifier
4. Adaboostclassifier
5. Logistic Regression
6. SVC

Outcome of the Lazy Predict is as follows:



Model	Accuracy	Balanced Accuracy	ROC AUC	F1 Score	Time Taken
LinearDiscriminantAnalysis	0.80	0.69	0.69	0.82	0.77
XGBClassifier	0.89	0.66	0.66	0.87	5.33
AdaBoostClassifier	0.85	0.64	0.64	0.84	0.88
LGBMClassifier	0.87	0.64	0.64	0.85	0.27
Perceptron	0.86	0.63	0.63	0.84	0.16
PassiveAggressiveClassifier	0.85	0.62	0.62	0.83	0.18
LinearSVC	0.85	0.61	0.61	0.83	3.05
GaussianNB	0.81	0.60	0.60	0.81	0.12
NearestCentroid	0.87	0.58	0.58	0.84	0.14
DecisionTreeClassifier	0.81	0.55	0.55	0.79	0.22
ExtraTreeClassifier	0.84	0.54	0.54	0.80	0.11
LogisticRegression	0.86	0.54	0.54	0.81	0.20
RidgeClassifier	0.85	0.54	0.54	0.81	0.19
RidgeClassifierCV	0.85	0.54	0.54	0.81	0.32
BaggingClassifier	0.84	0.53	0.53	0.80	0.62
ExtraTreesClassifier	0.86	0.52	0.52	0.80	0.80
QuadraticDiscriminantAnalysis	0.15	0.50	0.50	0.05	0.55

Pipelines

Pipelines have become more popular in data science, with anything from basic data pipelines to complicated machine learning pipelines available. A pipeline's main goal is to make data analytics and machine learning operations more efficient. Let's go a bit deeper now. (https://www.datarobot.com/blog/what-a-machine-learning-pipeline-is-and-why-its-important/)

An ML pipeline is a way of automating the machine learning workflow by allowing data to be converted and correlated into a model that can then be examined to provide outputs, according to one description. The process of feeding data into the ML model is totally automated using this form of ML pipeline (*What Is a Machine Learning Pipeline? | DataRobot Blog*, 2022).

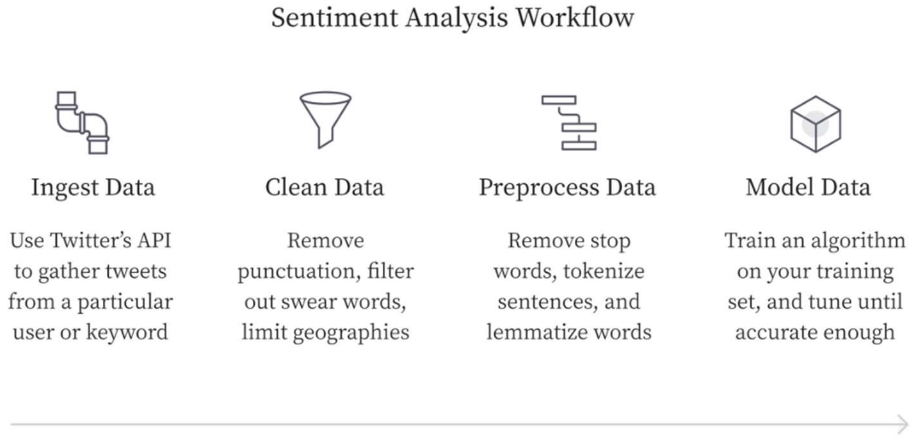

Sentiment Analysis Workflow
Ingest Data
Use Twitter's API to gather tweets from a particular user or keyword
Clean Data
Remove punctuation, filter out swear words, limit geographies
Preprocess Data
Remove stop words, tokenize sentences, and lemmatize words
Model Data
Train an algorithm on your training set, and tune until accurate enough

Figure: Sentiment Analysis Workflow(*What Is a Machine Learning Pipeline? | DataRobot Blog*, 2022)

(*What Is a Machine Learning Pipeline? | DataRobot Blog*, 2022)

All of these tasks would be executed in a monolith in a traditional system design. This implies that the data will be extracted, cleaned and prepared, modeled, and deployed using the same script. Because machine learning models often include significantly less code than conventional software applications, keeping all assets in one location makes sense. Each step of the workflow is abstracted into its own service using the ML pipeline. Then, whenever you create a new process, you can pick and choose the pieces you need and place them where you need them, while any modifications to that service are done at a higher level.

Natural language processing tasks sometimes require many processes that may be repeated. As an example, consider the following Twitter sentiment analysis pipeline. In the below image we can see the workflow of sentiment analysis.

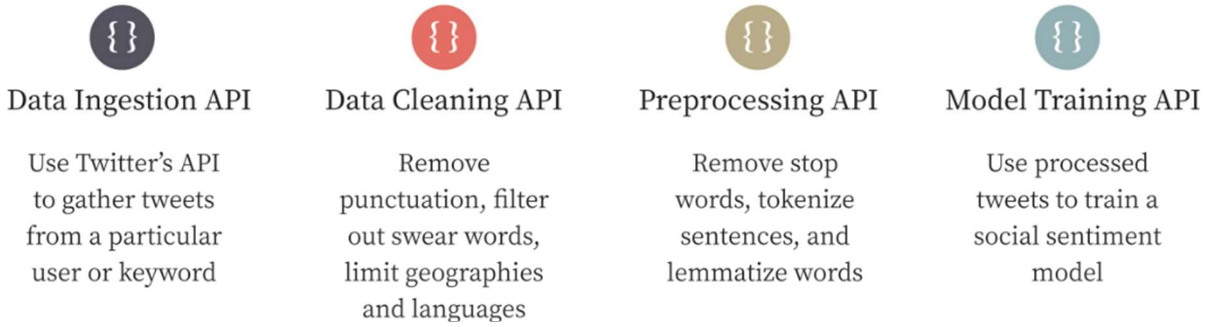

Data Ingestion API
Use Twitter's API to gather tweets from a particular user or keyword
Data Cleaning API
Remove punctuation, filter out swear words, limit geographies and languages
Preprocessing API
Remove stop words, tokenize sentences, and lemmatize words
Model Training API
Use processed tweets to train a social sentiment model

Figure: API Workflow(*What Is a Machine Learning Pipeline? | DataRobot Blog*, 2022).

This workflow involves ingesting data from Twitter, cleaning it for punctuation and whitespace, tokenizing and lemmatizing it, and then sending it via a sentiment analysis algorithm to classify it. At first, keeping all of these functions together makes sense, but as we performed additional studies on this dataset, we wanted to modularize the workflow. For comparison, we can see how the analysis would go without the pipeline and with the pipeline with monolithic structures.

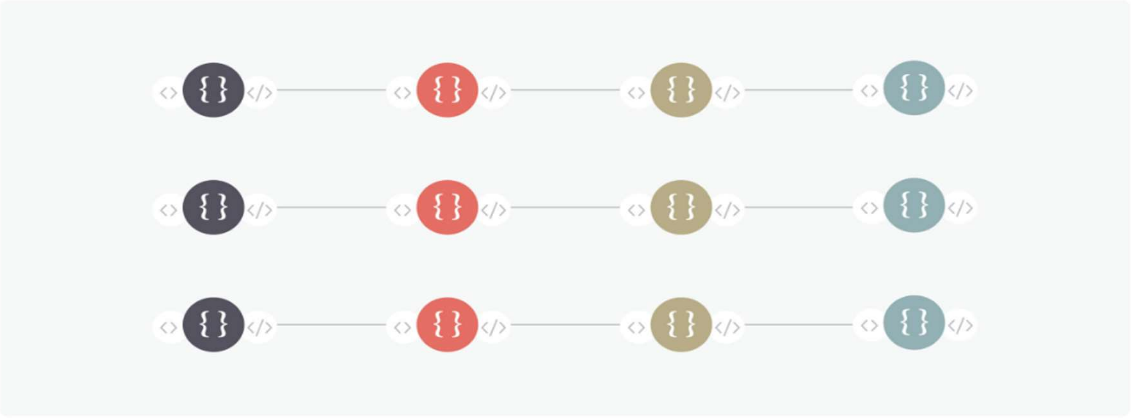


Figure: API Workflow without Pipeline (*What Is a Machine Learning Pipeline? | DataRobot Blog*, 2022).

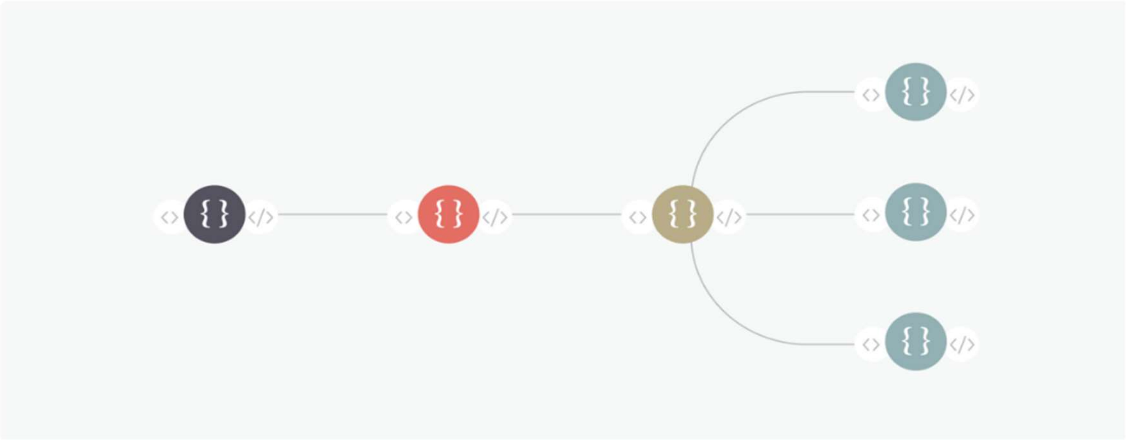


Figure: API Workflow with Pipeline (*What Is a Machine Learning Pipeline? | DataRobot Blog*, 2022).

The above images show that It's simple to change the algorithms, alter the cleaning or preprocessing processes, or scrape tweets from a different user using this architecture without interrupting the rest of your workflow. There will be no copying and pasting of changes across iterations, and the structure will function more smoothly with fewer overall elements.

In the case of the Twitter data, we have used Naive Bayes as the base model and ended up getting an accuracy of 86%. To create a better model we constructed a pipeline using six models namely, Decision Tree Classifier, Logistic Regression, Random Forest Classifier, Ridge Classifier, SVC, and AdaBoost classifier models. In pipeline testing, for the Twitter model, we found that the Ridge Classifier model outperformed the other models with an accuracy of 89 %.

Below we can see how the models performed in a graphical representation.

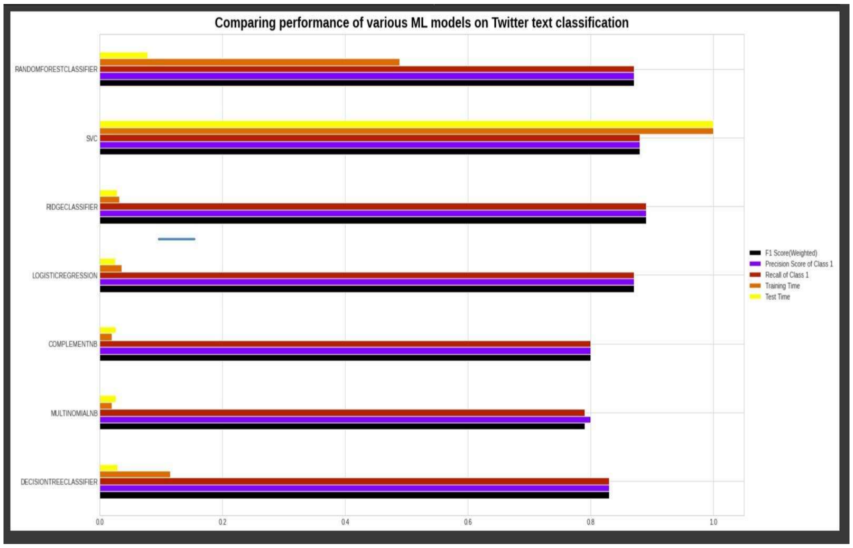


Fig: Comparing the performance of models on Twitter Data.

However, The data was also put through its paces using a basic classification model, the Logistic Regression model, which yielded an accuracy of 85 percent. When compared to other models, however, this accuracy was not judged the best option. To select the best fitting model for our data, a pipeline of 6 classification models was developed, including Decision Tree Classifier, Logistic Regression, Random Forest Classifier, Ridge Classifier, SVC, and AdaBoost classifier models. The Decision Tree Classifier model outperformed the other models in pipeline testing.

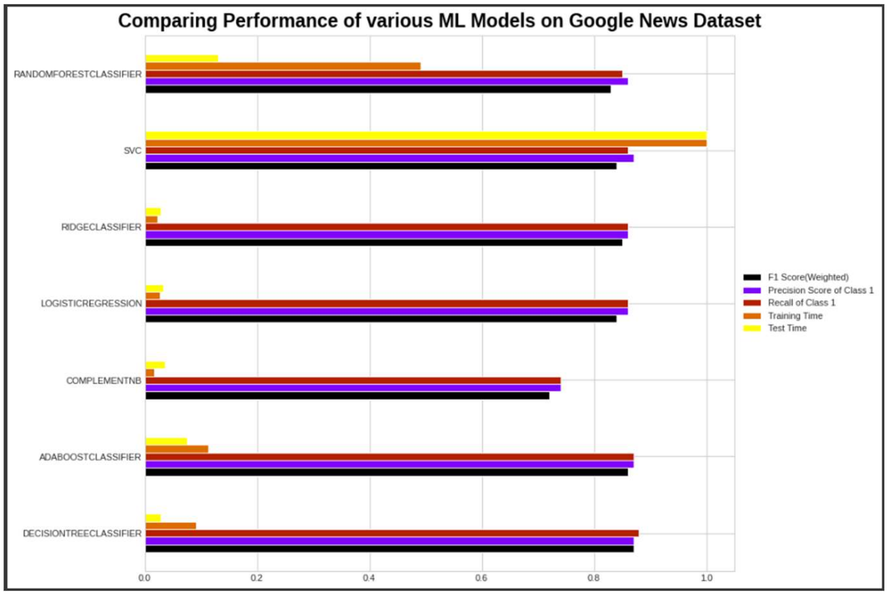


Fig: Comparing the performance of models on Google News Data.

To get a better understanding of the deep learning approach, we used TensorFlow and Keras modeling approaches to train the models. We trained the model across ten epochs to achieve an accuracy of 86%. When compared to the accuracy provided by pipeline models, the deep learning approach's accuracy fell short.

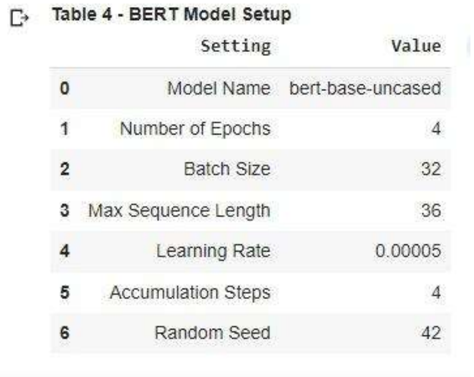
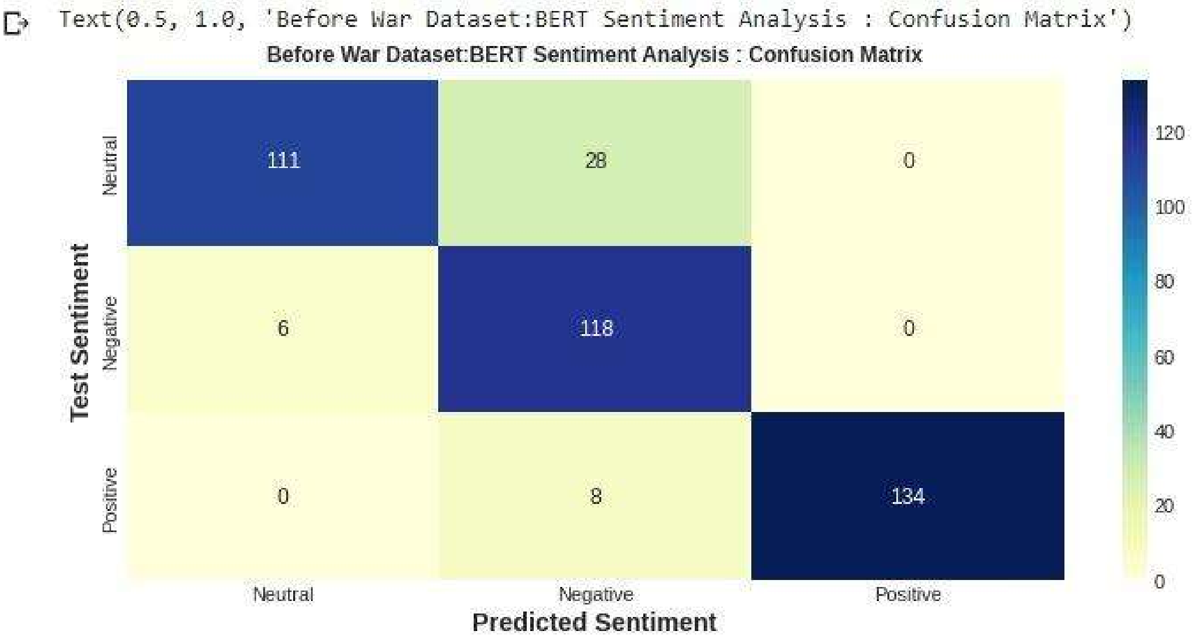


Table 4 - BERT Model Setup

	Setting	Value
0	Model Name	bert-base-uncased
1	Number of Epochs	4
2	Batch Size	32
3	Max Sequence Length	36
4	Learning Rate	0.00005
5	Accumulation Steps	4
6	Random Seed	42

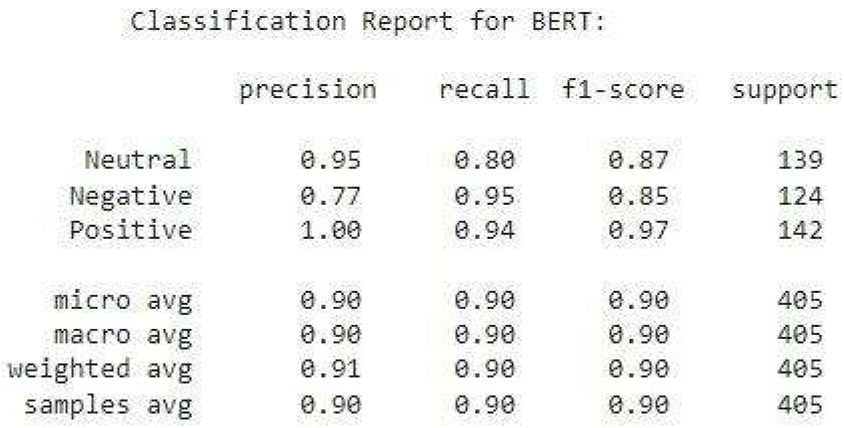
Fig: BERT model Setup.

Finally, we applied transformers to train the model in order to increase its accuracy. We used the BERT model for the transformer training model, which resulted in a 93 per cent accuracy, which was the finest fit.



```
Text(0.5, 1.0, 'Before War Dataset:BERT Sentiment Analysis : Confusion Matrix')
```

FIG: Confusion matrix of BERT model



```
Classification Report for BERT:

               precision    recall  f1-score   support

      Neutral       0.95      0.80      0.87       139
     Negative       0.77      0.95      0.85       124
     Positive       1.00      0.94      0.97       142

    micro avg       0.90      0.90      0.90       405
    macro avg       0.90      0.90      0.90       405
 weighted avg       0.91      0.90      0.90       405
  samples avg       0.90      0.90      0.90       405
```

Fig: Classification Report of BERT model

Conclusion

As a part of the invasion sentiment analysis we have tried different approaches as follows:

1. After the data processing and labeling of all three datasets we found that:
   a) In the before war dataset the neutral and negative sentiment counts were more.
   b) Following the same in during war dataset, we found that during the war the negative count was far more than the positive and neutral.
   c) In the combined dataset the neutral count was more than negative and positive. And as per the study these countries were neutral about the war due to their business relations with Russia.
   d) In the case of the Twitter majority of tweets had neutral sentiment followed by Negative with Positive being the last.
2. As Twitter data and Google News include users from all over the world who are not directly affected by the war, as a result, a high number of tweets fall under the neutral category.
3. After applying the pipeline we found of 6 models we found that the Decision Tree Classifier outperformed every other model with an accuracy of more than 85% in all three datasets for Google News and for Twitter the Ridge Classifier outperformed all.
4. To further analyze, we delved into the deep learning approach and tried to find out the accuracy of the sentiment through Tensorflow and Keras but couldn't find a better accuracy for the Google News Data.

5. At last, we applied the transformer approach for the sentiment analysis to a section of our data. Surprisingly, we got the best results using the transformer. We applied the BERT model for model training and got an accuracy of more than 93% with 4 epochs which were way better than the models we used in the pipeline and found the best accuracy with the ridge itself in the case of Twitter data.

Work Cited:

Ames, E., & Kononenko, K. (1959). Ukraine and Russia: A History of the Economic Relations between Ukraine and Russia (1654–1917). American Slavic and East European Review, 18(2), 258. https://doi.org/10.2307/3001374

Das, T. K. (2022). Russia-Ukraine War: Trust and Distrust. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.4060599

Wikipedia contributors. (2022, May 13). Russo-Ukrainian War. Wikipedia. https://en.wikipedia.org/wiki/Russo-Ukrainian_War

Fung, L. H., & M. Belaidan, S. L. (2021). Sentiment Analysis in Online Products Reviews Using Machine Learning. Webology, 18(SI05), 914–928. https://doi.org/10.14704/web/v18si05/web18271

Sentiment Analysis by using deep learning and Machine learning Techniques: A Review. (2021). International Journal of Advanced Trends in Computer Science and Engineering, 10(2), 754–761. https://doi.org/10.30534/ijatcse/2021/421022021

A Survey on Sentiment Analysis and Its Challenges using Machine Learning Algorithms. (2020). Journal of Xidian University, 14(8). https://doi.org/10.37896/jxu14.8/119

What Is Twitter & How Does It Work? (2021, August 30). Lifewire. https://www.lifewire.com/what-exactly-is-twitter-2483331

Iana Sabatovych. Do social media create revolutions? using twitter sentiment analysis for predicting the maidanrevolution in ukraine. Global Media and Communication, 15:275–283, 12 2019

Real Python. (2021, March 6). How to Make a Twitter Bot in Python With Tweepy. How to Make a Twitter Bot in Python With Tweepy. https://realpython.com/twitter-bot-python-

tweepy/#:%7E:text=Tweepy%20is%20an%20open%20source,Data%20encoding%20and%20decoding

What is a Machine Learning Pipeline? | DataRobot Blog. (2022, March 24). DataRobot AICloud. https://www.datarobot.com/blog/what-a-machine-learning-pipeline-is-and-why-its-important/

Dr.K, U. P. K., & Dr.M, K. (2020). Performance Analysis of Naïve Bayes Correlation Models in Machine Learning. International Journal of Psychosocial Rehabilitation, 24(04), 1153–1157. https://doi.org/10.37200/ijpr/v24i4/pr201088

Smetanin, S. (2020). The applications of sentiment analysis for Russian language texts: Current challenges and future perspectives. IEEE Access, 8, 110693-110719.

Efstathios Polyzos. Escalating tension and the war in ukraine: Evidence using impulse response functions oneconomic indicators and twitter sentiment. SSRN Electronic Journal, 2022.