Russia Ukrain Invasion Sentiment Analysis on Twitter Data

Data 602: Final Project

Russia invaded Ukraine 24th Feburary'2022. In this notebook we have tried to analye the sentiment of the people around the world by focusing on the news which is broadcasted on this subject across the globe. The project proceeds with the flow mentioned below:

1. Installing and Importing the required libraries.
2. Data Cleaning and Data Wrangling.
3. Basic analysis on the data.
4. Sentiment Analysis
5. Model training through naive Bayes and pipeline.
6. Conclusion.
7. Future Work.

Installing relevant libraries

```
!pip install textblob
```

```
Requirement already satisfied: textblob in /usr/local/lib/python3.7/dist-packages (0.15
Requirement already satisfied: nltk>=3.1 in /usr/local/lib/python3.7/dist-packages (fron
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from nltk>
```

```
!pip install transformers
```

```
Collecting transformers
  Downloading transformers-4.19.0-py3-none-any.whl (4.2 MB)
     |████████████████████████████████| 4.2 MB 5.1 MB/s
Collecting pyyaml>=5.1
  Downloading PyYAML-6.0-cp37-cp37m-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_1
     |████████████████████████████████| 596 kB 60.5 MB/s
Collecting huggingface-hub<1.0,>=0.1.0
  Downloading huggingface_hub-0.6.0-py3-none-any.whl (84 kB)
     |████████████████████████████████| 84 kB 2.0 MB/s
Collecting tokenizers!=0.11.3,<0.13,>=0.11.1
  Downloading tokenizers-0.12.1-cp37-cp37m-manylinux_2_12_x86_64.manylinux2010_x86_64.wh
     |████████████████████████████████| 6.6 MB 40.6 MB/s
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.7/dist-packages (fro
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: filelock in /usr/local/lib/python3.7/dist-packages (from
```

```
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.7/di
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.7/dist
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (t
Installing collected packages: pyyaml, tokenizers, huggingface-hub, transformers
  Attempting uninstall: pyyaml
    Found existing installation: PyYAML 3.13
    Uninstalling PyYAML-3.13:
      Successfully uninstalled PyYAML-3.13
Successfully installed huggingface-hub-0.6.0 pyyaml-6.0 tokenizers-0.12.1 transformers-4
```

```
!pip install tensorflow
```

```
Requirement already satisfied: tensorflow in /usr/local/lib/python3.7/dist-packages (2.8
Requirement already satisfied: keras-preprocessing>=1.1.1 in /usr/local/lib/python3.7/di
Requirement already satisfied: protobuf>=3.9.2 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/py
Requirement already satisfied: absl-py>=0.4.0 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: keras<2.9,>=2.8.0rc0 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-packages (fro
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.7/dist-packag
Collecting tf-estimator-nightly==2.8.0.dev2021122109
  Downloading tf_estimator_nightly-2.8.0.dev2021122109-py2.py3-none-any.whl (462 kB)
     |████████████████████████████████| 462 kB 4.8 MB/s
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.7/dist
Requirement already satisfied: libclang>=9.0.1 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: h5py>=2.9.0 in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: flatbuffers>=1.12 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: gast>=0.2.1 in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: tensorboard<2.9,>=2.8 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: cached-property in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /usr/local/lib/python3.7
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /usr/local/lib/python
Requirement already satisfied: werkzeug>=0.11.15 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in /usr/local/lib/p
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.7/dist-packages (
```

```
Requirement already satisfied: cachetools<5.0,>=2.0.0 in /usr/local/lib/python3.7/dist-p
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.7/dist
Requirement already satisfied: importlib-metadata>=4.4 in /usr/local/lib/python3.7/dist-
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.7/dist-packages
Installing collected packages: tf-estimator-nightly
Successfully installed tf-estimator-nightly-2.8.0.dev2021122109
```

```
!pip install snorkel
```

```
Collecting snorkel
  Downloading snorkel-0.9.8-py3-none-any.whl (103 kB)
     |████████████████████████████████| 103 kB 4.4 MB/s
Requirement already satisfied: pandas<2.0.0,>=1.0.0 in /usr/local/lib/pyth
Requirement already satisfied: torch<2.0.0,>=1.2.0 in /usr/local/lib/pythc
Collecting scikit-learn<0.25.0,>=0.20.2
  Downloading scikit_learn-0.24.2-cp37-cp37m-manylinux2010_x86_64.whl (22.
     |████████████████████████████████| 22.3 MB 3.4 MB/s
Requirement already satisfied: scipy<2.0.0,>=1.2.0 in /usr/local/lib/pythc
Requirement already satisfied: tqdm<5.0.0,>=4.33.0 in /usr/local/lib/pythc
Collecting munkres>=1.0.6
  Downloading munkres-1.1.4-py2.py3-none-any.whl (7.0 kB)
Requirement already satisfied: networkx<2.7,>=2.2 in /usr/local/lib/pythor
Collecting tensorboard<2.7.0,>=2.0.0
  Downloading tensorboard-2.6.0-py3-none-any.whl (5.6 MB)
     |████████████████████████████████| 5.6 MB 41.6 MB/s
Collecting numpy<1.20.0,>=1.16.5
  Downloading numpy-1.19.5-cp37-cp37m-manylinux2010_x86_64.whl (14.8 MB)
     |████████████████████████████████| 14.8 MB 36.8 MB/s
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/di
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/py
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-p
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.7/di
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/pyth
Requirement already satisfied: protobuf>=3.6.0 in /usr/local/lib/python3.7
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in /u
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /usr/lc
Requirement already satisfied: wheel>=0.26 in /usr/local/lib/python3.7/dis
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/pythc
Requirement already satisfied: werkzeug>=0.11.15 in /usr/local/lib/python3
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.7
Requirement already satisfied: grpcio>=1.24.3 in /usr/local/lib/python3.7/
Requirement already satisfied: google-auth<2,>=1.6.3 in /usr/local/lib/pyt
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /usr/local
Requirement already satisfied: setuptools>=41.0.0 in /usr/local/lib/pythor
```

```
!pip install spacy
```

```
Requirement already satisfied: spacy in /usr/local/lib/python3.7/dist-packages (2.2.4)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: wasabi<1.1.0,>=0.4.0 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: srsly<1.1.0,>=1.0.2 in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: thinc==7.4.0 in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: numpy>=1.15.0 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: blis<0.5.0,>=0.4.0 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: plac<1.2.0,>=0.9.6 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-packages (fro
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.7/dist-
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.7/dis
Requirement already satisfied: catalogue<1.1.0,>=0.0.7 in /usr/local/lib/python3.7/dist-
Requirement already satisfied: importlib-metadata>=0.20 in /usr/local/lib/python3.7/dist
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: typing-extensions>=3.6.4 in /usr/local/lib/python3.7/dist
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib
```

```
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (f
```

```
!pip install tweepy stylecloud -q
```

```
     |████████████████████████████████| 262 kB 5.1 MB/s
     |████████████████████████████████| 161 kB 39.9 MB/s
     |████████████████████████████████| 87 kB 4.5 MB/s
     |████████████████████████████████| 87 kB 3.6 MB/s
  Building wheel for stylecloud (setup.py) ... done
  Building wheel for fire (setup.py) ... done
  Building wheel for tinycss (setup.py) ... done
```

```
!pip install plotly
```

```
Requirement already satisfied: plotly in /usr/local/lib/python3.7/dist-packages (5.5.0)
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from plotl
```

```
!pip install pyyaml==5.4.1
```

```
Collecting pyyaml==5.4.1
  Downloading PyYAML-5.4.1-cp37-cp37m-manylinux1_x86_64.whl (636 kB)
     |████████████████████████████████| 636 kB 5.1 MB/s
Installing collected packages: pyyaml
  Attempting uninstall: pyyaml
    Found existing installation: PyYAML 6.0
    Uninstalling PyYAML-6.0:
      Successfully uninstalled PyYAML-6.0
Successfully installed pyyaml-5.4.1
```

```
#Snorkel
from snorkel.labeling import LabelingFunction
import re
from snorkel.preprocess import preprocessor
from textblob import TextBlob
from snorkel.labeling import PandasLFApplier
from snorkel.labeling.model import LabelModel
from snorkel.labeling import LFAnalysis
from snorkel.labeling import filter_unlabeled_dataframe
from snorkel.labeling import labeling_function
#NLP packages
import spacy
from nltk.corpus import stopwords
import string
import nltk
import nltk.tokenize
punc = string.punctuation
```

```
nltk.download('stopwords')
stop_words = set(stopwords.words('english'))
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

```
#general purpose packages
import numpy as np
import pandas as pd
import tensorflow as tf
import matplotlib.pyplot as plt
import seaborn as sns
import plotly
from textblob import TextBlob
import stylecloud


#data processing
import re, string
#import emoji
import nltk

from sklearn import preprocessing
from imblearn.over_sampling import RandomOverSampler
from sklearn.model_selection import train_test_split


#Naive Bayes
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.naive_bayes import MultinomialNB

#transformers
from transformers import BertTokenizerFast
from transformers import TFBertModel

# #keras
# import tensorflow as tf
# from tensorflow import keras


#metrics
from sklearn.metrics import accuracy_score, f1_score
from sklearn.metrics import classification_report, confusion_matrix

#set seed for reproducibility
seed=42

#set style for plots
```

```
sns.set_style("whitegrid")
sns.despine()
plt.style.use("seaborn-whitegrid")
plt.rc("figure", autolayout=True)
plt.rc("axes", labelweight="bold", labelsize="large", titleweight="bold", titlepad=10)

df1=pd.read_csv('https://raw.githubusercontent.com/ajitjadhav10/UMBC/main/DATA%20602/Project/
df2=pd.read_csv('https://raw.githubusercontent.com/ajitjadhav10/UMBC/main/DATA%20602/Project/
df3=pd.read_csv('https://raw.githubusercontent.com/ajitjadhav10/UMBC/main/DATA%20602/Project/
df4=pd.read_csv('https://raw.githubusercontent.com/ajitjadhav10/UMBC/main/DATA%20602/Project/
df5=pd.read_csv('https://raw.githubusercontent.com/ajitjadhav10/UMBC/main/DATA%20602/Project/
df6=pd.read_csv('https://raw.githubusercontent.com/ajitjadhav10/UMBC/main/DATA%20602/Project/
df7=pd.read_csv('https://raw.githubusercontent.com/ajitjadhav10/UMBC/main/DATA%20602/Project/
df8=pd.read_csv('https://raw.githubusercontent.com/ajitjadhav10/UMBC/main/DATA%20602/Project/
df9=pd.read_csv('https://raw.githubusercontent.com/ajitjadhav10/UMBC/main/DATA%20602/Project/
df10=pd.read_csv('https://raw.githubusercontent.com/ajitjadhav10/UMBC/main/DATA%20602/Project
df11=pd.read_csv('https://raw.githubusercontent.com/ajitjadhav10/UMBC/main/DATA%20602/Project
df12=pd.read_csv('https://raw.githubusercontent.com/ajitjadhav10/UMBC/main/DATA%20602/Project
df13=pd.read_csv('https://raw.githubusercontent.com/ajitjadhav10/UMBC/main/DATA%20602/Project
df14=pd.read_csv('https://raw.githubusercontent.com/ajitjadhav10/UMBC/main/DATA%20602/Project
df15=pd.read_csv('https://raw.githubusercontent.com/ajitjadhav10/UMBC/main/DATA%20602/Project


df_new=pd.concat([df1,df2,df3,df4,df5,df6,df7,df8,df9,df10,df11,df12,df13,df14,df15],ignore_i
df_new.head()
```

Unnamed: 0	Unnamed: 0.1	userid	username	acctdesc	location	fol

Exploratory data analysis

```
#Describing the data

df_new.describe()
```

	Unnamed: 0	Unnamed: 0.1	userid	following	followers	totaltwe
count	364875.000000	364875.000000	3.648750e+05	364875.000000	3.648750e+05	3.648750e
mean	182437.000000	182437.000000	6.400115e+17	1885.067059	1.917747e+04	6.201348e
std	105330.484073	105330.484073	6.464213e+17	6485.145732	3.694885e+05	1.554691e
min	0.000000	0.000000	7.670000e+02	0.000000	0.000000e+00	0.000000e
25%	91218.500000	91218.500000	4.894283e+08	159.000000	7.100000e+01	2.862000e
50%	182437.000000	182437.000000	7.444840e+17	567.000000	3.640000e+02	1.380400e
75%	273655.500000	273655.500000	1.326440e+18	1837.000000	1.567000e+03	5.575200e
max	364874.000000	364874.000000	1.510040e+18	483344.000000	1.695393e+07	4.035049e

```
#Printing the count of columns and rows in the dataset

print('Count of columns in the dataset is:  ', len(df_new.columns))
print('Count of rows in the dataset is:  ', len(df_new))
```

```
Count of columns in the dataset is:   19
Count of rows in the dataset is:   364875
```

```
df_new.isnull().sum()
```

```
Unnamed: 0             0
Unnamed: 0.1           0
userid                 0
username               0
acctdesc           78444
location          151942
following              0
followers              0
totaltweets            0
usercreatedts          0
tweetid                0
tweetcreatedts         0
retweetcount           0
```

```
tweet                 0
hashtags              0
language              0
coordinates      364778
favorite_count        0
extractedts           0
dtype: int64
```

```
df_new.columns
```

```
Index(['Unnamed: 0', 'Unnamed: 0.1', 'userid', 'username', 'acctdesc',
       'location', 'following', 'followers', 'totaltweets', 'usercreatedts',
       'tweetid', 'tweetcreatedts', 'retweetcount', 'tweet', 'hashtags',
       'language', 'coordinates', 'favorite_count', 'extractedts'],
      dtype='object')
```

```
df_new.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 364875 entries, 0 to 364874
Data columns (total 19 columns):
 #   Column          Non-Null Count   Dtype
---  ------          --------------   -----
 0   Unnamed: 0      364875 non-null  int64
 1   Unnamed: 0.1    364875 non-null  int64
 2   userid          364875 non-null  float64
 3   username        364875 non-null  object
 4   acctdesc        286431 non-null  object
 5   location        212933 non-null  object
 6   following       364875 non-null  int64
 7   followers       364875 non-null  int64
 8   totaltweets     364875 non-null  int64
 9   usercreatedts   364875 non-null  object
 10  tweetid         364875 non-null  float64
 11  tweetcreatedts  364875 non-null  object
 12  retweetcount    364875 non-null  int64
 13  tweet           364875 non-null  object
 14  hashtags        364875 non-null  object
 15  language        364875 non-null  object
 16  coordinates     97 non-null      object
 17  favorite_count  364875 non-null  int64
 18  extractedts     364875 non-null  object
dtypes: float64(2), int64(7), object(10)
memory usage: 52.9+ MB
```

Count of tweets according to language

```
df_new_1=pd.DataFrame(df_new.language.value_counts()).reset_index()
```

```
df_new_1.head()
```

	index	language
0	en	254626
1	fr	18647
2	de	16446
3	it	15877
4	und	15613

```
df_top_lang=df_new_1.head(10)
```

Plotting the top 10 languages used on twitter

```
import plotly.express as px
fig_1 = px.bar(df_top_lang,
               x="index",
               y="language",
               title="Top 10 languages with most tweets",
               labels={"index":"Language","language":"Language"},
               color="index",
               hover_data=['language'],
               height=400
              ).update_xaxes(categoryorder="total descending")
fig_1.show()
```

```
/usr/local/lib/python3.7/dist-packages/distributed/config.py:20: YAMLLoadWarning: callir
  defaults = yaml.load(f)
```

We can observe from the above graph that english(en) is the predominant language followed by french(fr) and German(de) in second and third place respectively

Printing the top 10 retweeted tweets

```
df_top_retweet=df_new.sort_values(by=['retweetcount'],ascending=False)
```

```
df_top_retweet=df_top_retweet[['username','tweet','retweetcount']]
df_top_retweet.head(10)
```

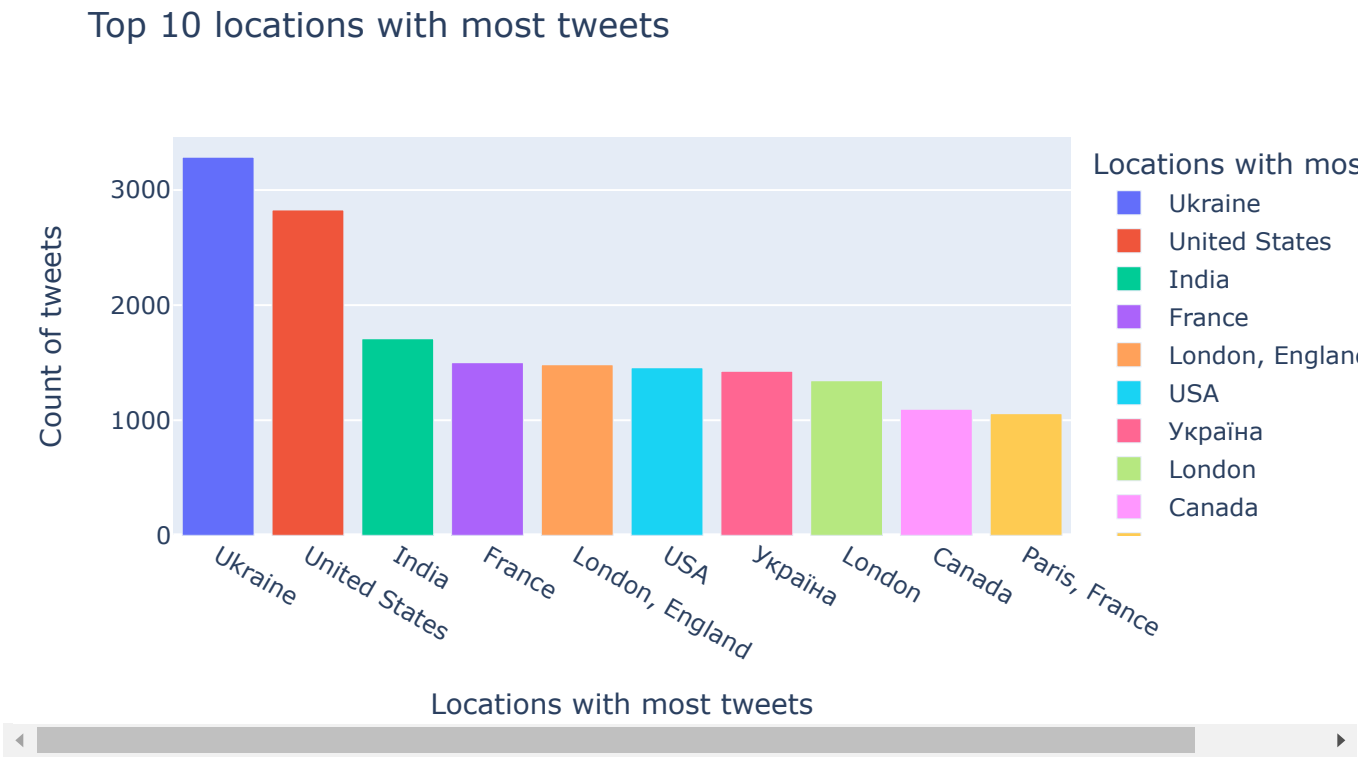
	username	tweet	retweetcount
35910	KathyBrownKathy	.@ZelenskyyUa's tv address to the Russian (!) ...	147055
49976	TriciaFoster	.@ZelenskyyUa's tv address to the Russian (!) ...	147053
111599	FranklynStarr	.@ZelenskyyUa's tv address to the Russian (!) ...	147052
226213	sunnnnnohhh	.@ZelenskyyUa's tv address to the Russian (!) ...	147039
230153	GTFund	.@ZelenskyyUa's tv address to the Russian (!) ...	147038
337976	MaartenKramer	.@ZelenskyyUa's tv address to the Russian (!) ...	147029

Printing the top 10 countries with most tweets

```
df_location=pd.DataFrame(df_new.location.value_counts()).reset_index()

df_location=df_location.head(10)
df_location.head()
```

	index	location
0	Ukraine	3285
1	United States	2827
2	India	1708
3	France	1500

```
fig_2 = px.bar(df_location,
               x="index",
               y="location",
               title="Top 10 locations with most tweets",
               labels={"index":"Locations with most tweets","location":"Count of tweets"},
               color="index",
               hover_data=['location'],
               height=400
              ).update_xaxes(categoryorder="total descending")
fig_2.show()
```



From the above plot we can see that naturally Ukraine is the top location followed by USA, India and France

Plotting the word cloud of the terms that are frequently used in tweets related to Russia invasion of Ukraine.

```
df_new['tweet'].to_csv('tweets.csv',index=False)
```

```
stylecloud.gen_stylecloud(file_path='tweets.csv',
                          icon_name='fab fa-twitter',
                          palette='colorbrewer.qualitative.Paired_3',
                          background_color='white',
                          gradient='horizontal',
                          stopwords=True,
                          custom_stopwords=['philipvollet','RT','THE','IS','WITH','ON','THIS'
                         )
```

```
from IPython.display import Image
Image('stylecloud.png')
```

```
df_new_en=df_new.loc[df_new['language']=='en',:]
df_new_en.head()
```

	Unnamed: 0	Unnamed: 0.1	userid	username	acctdesc	
0	0	0	1.688277e+07	Yaniela	Animal lover, supports those who fight injusti...	
1	1	1	3.205296e+09	gregffff	NaN	
2	2	2	1.235940e+18	ThanapornThon17	เล่นไวโอลิน\nพูดภาษาจีน	
3	3	3	1.347990e+18	I_Protest_2021	01000001 01101110 01101111 01101110 01111001 0...	Ir
					🌿 @Pickaw	

Cleaning the dataset for Sentiment Analysis

```
!pip install neattext
```

```
Collecting neattext
  Downloading neattext-0.1.3-py3-none-any.whl (114 kB)
     |████████████████████████████████| 114 kB 5.2 MB/s
Installing collected packages: neattext
Successfully installed neattext-0.1.3
```

```
import neattext.functions as nfx
```

```
dir(nfx)
```

```
['BTC_ADDRESS_REGEX',
 'CURRENCY_REGEX',
 'CURRENCY_SYMB_REGEX',
 'Counter',
 'DATE_REGEX',
 'EMAIL_REGEX',
 'EMOJI_REGEX',
 'HASTAG_REGEX',
 'MASTERCard_REGEX',
 'MD5_SHA_REGEX',
 'MOST_COMMON_PUNCT_REGEX',
 'NUMBERS_REGEX',
 'PHONE_REGEX',
 'PoBOX_REGEX',
 'SPECIAL_CHARACTERS_REGEX',
 'STOPWORDS',
 'STOPWORDS_de',
 'STOPWORDS_en',
 'STOPWORDS_es',
 'STOPWORDS_fr',
 'STOPWORDS_ru',
 'STOPWORDS_yo',
 'STREET_ADDRESS_REGEX',
 'TextFrame',
 'URL_PATTERN',
 'USER_HANDLES_REGEX',
 'VISACard_REGEX',
 '__builtins__',
 '__cached__',
 '__doc__',
 '__file__',
 '__generate_text',
 '__loader__',
 '__name__',
 '__numbers_dict',
 '__package__',
 '__spec__',
 '_lex_richness_herdan',
 '_lex_richness_maas_ttr',
 'clean_text',
```

```
 'defaultdict',
 'digit2words',
 'extract_btc_address',
 'extract_currencies',
 'extract_currency_symbols',
 'extract_dates',
 'extract_emails',
 'extract_emojis',
 'extract_hashtags',
 'extract_html_tags',
 'extract_mastercard_addr',
 'extract_md5sha',
 'extract_numbers',
 'extract_pattern',
 'extract_phone_numbers',
 'extract_postoffice_box',
 'extract_shortwords',
```

```
#Having a look at one of the tweets to understand what all things we need to clean out of the
```

```
df_new_en['tweet'].iloc[0]
```

```
' ⚡ The Ukrainian Air Force would like to address misinformation published
in multiple Western media outlets regarding the situation in the UA sky a
```

```
df_new_en['tweet'].apply(nfx.extract_hashtags)
```

```
0         [#ProtectUASky, #StopRussia, #UkraineUnderAttack]
1         [#russianinvasion., #StandWithUkraine, #Ukrai...
2          [#RussianUkrainianWar...Taiwan, #China, #Taiwan]
3                     [#Anonymous, #OpRussia, #DDoSecrets]
4                                           [#nft, #mint]
                                  ...
364866                                [#Bucha, #Russian]
364869    [#RussianUkrainianWar, #UkraineRussianWar, #Ru...
364871                                        [#Ukraine]
364872                                    [#SlavaUkraini]
364874                                        [#UKRAINE]
Name: tweet, Length: 254626, dtype: object
```

```
df_new_en['extracted_hashtags']=df_new_en['tweet'].apply(nfx.extract_hashtags)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
```

```
df_new_en[['extracted_hashtags','hashtags']]
```

	extracted_hashtags	hashtags
0	[#ProtectUASky, #StopRussia, #UkraineUnderAttack]	[]
1	[#russianinvasion., #StandWithUkraine, #Ukrai...	[{'text': 'russianinvasion', 'indices': [77, 9...
2	[#RussianUkrainianWar...Taiwan, #China, #Taiwan]	[{'text': 'RussianUkrainianWar', 'indices': [7...
3	[#Anonymous, #OpRussia, #DDoSecrets]	[{'text': 'Anonymous', 'indices': [25, 35]}]
4	[#nft, #mint]	[]
...	...	...
364866	[#Bucha, #Russian]	[{'text': 'Bucha', 'indices': [36, 42]}]
364869	[#RussianUkrainianWar, #UkraineRussianWar, #Ru...	[{'text': 'RussianUkrainianWar', 'indices': [0...

```
df_new_en['clean_tweet']=df_new_en['tweet'].apply(nfx.remove_hashtags)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
```

```
#We have removed the hashtags and we are comparing the original and cleaned tweets columns

df_new_en[['tweet','clean_tweet']]
```

	tweet	clean_tweet
0	⚡ The Ukrainian Air Force would like to address...	⚡ The Ukrainian Air Force would like to address...
1	Chernihiv oblast. Ukrainians welcome their lib...	Chernihiv oblast. Ukrainians welcome their lib...
2	America ᴜꜱ is preparing for something worse th...	America ᴜꜱ is preparing for something worse th...

```
df_new_en['clean_tweet'] =df_new_en['clean_tweet'].apply(lambda x:nfx.remove_userhandles(x))
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
```

```
#Now we can see that all user handles have been removed from the clean tweet column

df_new_en[['tweet','clean_tweet']]
```

	tweet	clean_tweet
0	⚡ The Ukrainian Air Force would like to address...	⚡ The Ukrainian Air Force would like to address...
1	Chernihiv oblast. Ukrainians welcome their lib...	Chernihiv oblast. Ukrainians welcome their lib...
2	America ᴜꜱ is preparing for something worse th...	America ᴜꜱ is preparing for something worse th...
3	JUST IN: #Anonymous has hacked & released ...	JUST IN: has hacked & released 62,000 em...
4	***PUBLIC MINT NOW LIVE***\n\nFor \n@billionai...	***PUBLIC MINT NOW LIVE***\n\nFor \n \n\nWin $...
...	...	...
364866	14-year-old Yura from #Bucha told how a Russia...	14-year-old Yura from told how a Russian sol...

#RussianUkrainianWar

```
#Removing multiple whitespaces
```

```
df_new_en['clean_tweet']=df_new_en['clean_tweet'].apply(nfx.remove_multiple_spaces)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user
```

```
#Removing urls

df_new_en['clean_tweet']=df_new_en['clean_tweet'].apply(nfx.remove_urls)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user
```

```
#removing punctuations
df_new_en['clean_tweet']=df_new_en['clean_tweet'].apply(nfx.remove_puncts)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user
```

```
df_new_en['clean_tweet']=df_new_en['clean_tweet'].apply(nfx.remove_emojis)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user
```

```
df_new_en['clean_tweet']=df_new_en['clean_tweet'].apply(nfx.remove_special_characters)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
```

```
df_new_en
```

	Unnamed: 0	Unnamed: 0.1	userid	username	acctde
0	0	0	1.688277e+07	Yaniela	Animal lov supports those w fight injust
1	1	1	3.205296e+09	gregffff	N
2	2	2	1.235940e+18	ThanapornThon17	เล่นไวโอลิน\n ภาษา

Sentiment Analysis

```
#defining constants to represent the class labels :positive, negative, and abstain
POSITIVE = 1
NEGATIVE = 0
ABSTAIN = -1
#define function which looks into the input words to represent a proper label
def keyword_lookup(x, keywords, label):
    if any(word in x.text.lower() for word in keywords):
        return label
    return ABSTAIN
#define function which assigns a correct label
def make_keyword_lf(keywords, label=POSITIVE):
    return LabelingFunction(
        name=f"keyword_{keywords[0]}",
        f=keyword_lookup,
        resources=dict(keywords=keywords, label=label))

#these two lists can be further extended
"""positive news might contain the following words' """
keyword_positive = make_keyword_lf(keywords=['boosts', 'great', 'develops', 'promising', 'amb
                                             'peace', 'party', 'hope', 'flourish', 'respect',
                                             'perfect', 'complete', 'assured' ])
"""negative news might contain the following words"""
keyword_negative = make_keyword_lf(keywords=['war','solidiers', 'turmoil', 'injur','trouble',
                                             'defeat', 'damage', 'dishonest', 'dead', 'fear',
                                             'fraud', 'dispute', 'destruction', 'battle', 'un
                                             'unhealthy', 'tensions','emergency', 'Accident',
```

```
                                                    'weaponizing', 'crisis', 'warships', 'pessimisti
                                                    'complicate', 'separatists'], label=NEGATIVE)
```

```
#set up a preprocessor function to determine polarity & subjectivity using textlob pretrained
@preprocessor(memoize=True)
def textblob_sentiment(x):
    scores = TextBlob(x.text)
    x.polarity = scores.sentiment.polarity
    x.subjectivity = scores.sentiment.subjectivity
    return x
#find polarity
@labeling_function(pre=[textblob_sentiment])
def textblob_polarity(x):
    return POSITIVE if x.polarity > 0.6 else ABSTAIN
#find subjectivity
@labeling_function(pre=[textblob_sentiment])
def textblob_subjectivity(x):
    return POSITIVE if x.subjectivity >= 0.5 else ABSTAIN
```

```
#conduct some data cleaning
df_new_en = df_new_en[['username','clean_tweet']]
df_new_en = df_new_en.rename(columns = {'clean_tweet': 'text'})
df_new_en['text'] = df_new_en['text'].astype(str)
df_new_en.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 254626 entries, 0 to 364874
Data columns (total 2 columns):
 #   Column    Non-Null Count   Dtype
---  ------    --------------   -----
 0   username  254626 non-null  object
 1   text      254626 non-null  object
dtypes: object(2)
memory usage: 5.8+ MB
```

```
#combine all the labeling functions
lfs = [keyword_positive, keyword_negative, textblob_polarity, textblob_subjectivity ]
#apply the lfs on the dataframe
applier = PandasLFApplier(lfs=lfs)
L_snorkel = applier.apply(df=df_new_en)
#apply the label model
label_model = LabelModel(cardinality=2, verbose=True)
#fit on the data
label_model.fit(L_snorkel)
#predict and create the labels
df_new_en["label"] = label_model.predict(L=L_snorkel)
```

```
100%|██████████| 254626/254626 [06:35<00:00, 643.27it/s]
INFO:root:Computing O...
INFO:root:Estimating \mu...
  0%|          | 0/100 [00:00<?, ?epoch/s]INFO:root:[0 epochs]: TRAIN:[loss=0.046]
  6%|▌         | 6/100 [00:00<00:01, 59.68epoch/s]INFO:root:[10 epochs]: TRAIN:[loss=0.6
```

```
INFO:root:[20 epochs]: TRAIN:[loss=0.001]
INFO:root:[30 epochs]: TRAIN:[loss=0.003]
INFO:root:[40 epochs]: TRAIN:[loss=0.002]
INFO:root:[50 epochs]: TRAIN:[loss=0.001]
INFO:root:[60 epochs]: TRAIN:[loss=0.001]
INFO:root:[70 epochs]: TRAIN:[loss=0.001]
INFO:root:[80 epochs]: TRAIN:[loss=0.001]
INFO:root:[90 epochs]: TRAIN:[loss=0.001]
100%|██████████| 100/100 [00:00<00:00, 462.13epoch/s]
INFO:root:Finished Training
```

```
#Filtering out unlabeled data points
df_new_en= df_new_en.loc[df_new_en.label.isin([0,1,-1]), :]
#find the label counts
df_new_en['label'].value_counts()
```

```
-1    120094
 0     79143
 1     55389
Name: label, dtype: int64
```

```
df_new_en.head(20)
```

	username	text	label
0	Yaniela	The Ukrainian Air Force would like to address ...	-1
1	gregffff	Chernihiv oblast Ukrainians welcome their libe...	1
2	ThanapornThon17	America is preparing for something worse than...	0
3	I_Protest_2021	JUST IN has hacked amp released 62000 emails f...	-1

```
df_new_en['label'].value_counts()
```

```
-1    120094
 0     79143
 1     55389
Name: label, dtype: int64
```

7	livemint	Indias purchase of discounted crude oil and pu...	1

```
df_new_en['sentiment'] = df_new_en['label'].map({-1:'Neutral',0:'Negative',1:'Positive'})
df_new_en
```

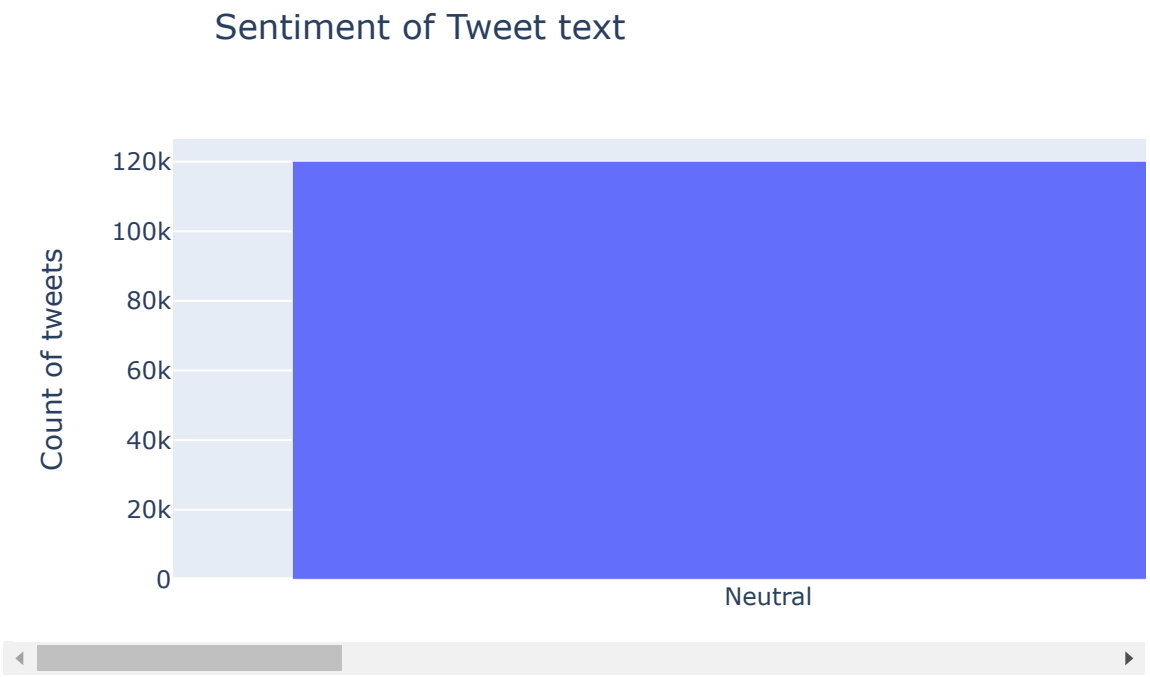
	username	text	label	sentiment
0	Yaniela	The Ukrainian Air Force would like to address ...	-1	Neutral
1	gregffff	Chernihiv oblast Ukrainians welcome their libe...	1	Positive
2	ThanapornThon17	America is preparing for something worse than...	0	Negative
3	I_Protest_2021	JUST IN has hacked amp released 62000 emails f...	-1	Neutral
4	Marsh_Win_01	PUBLIC MINT NOW LIVE For Win 100000 during pub...	1	Positive
...	...	...	...	...
364866	KatCapps	14yearold Yura from told how a Russian soldier...	0	Negative

I saw the video 3 months ago or am

```
df_sentiment=pd.DataFrame(df_new_en.sentiment.value_counts()).reset_index()

df_sentiment.head()
```

index	sentiment

```
fig_3 = px.bar(df_sentiment,
               x="index",
               y="sentiment",
               title="Sentiment of Tweet text",
               labels={"index":"Sentiment","sentiment":"Count of tweets"},
               color="index",
               hover_data=['sentiment'],
               height=400
              ).update_xaxes(categoryorder="total descending")
fig_3.show()
```




```
def strip_all_entities(text):
    text = text.replace('\r', '').replace('\n', ' ').replace('\n', ' ').lower()
    text = re.sub(r"(?:\@|https?\://)\S+", "", text)
    text = re.sub(r'[^\x00-\x7f]',r'', text)
    banned_list= string.punctuation + 'Ã'+'±'+'ã'+'¼'+'â'+'»'+'§'
    table = str.maketrans('', '', banned_list)
    text = text.translate(table)
    return text


#Filter special characters such as & and $ present in some words
def filter_chars(a):
    sent = []
    for word in a.split(' '):
        if ('$' in word) | ('&' in word):
```

```
            sent.append('')
        else:
            sent.append(word)
    return ' '.join(sent)
```

```
new_text = []
for t in df_new_en.text:
    new_text.append(filter_chars(strip_all_entities(t)))
```

```
df_new_en['text'] = new_text
```

```
df_new_en.head()
```

	username	text	label	sentiment
0	Yaniela	the ukrainian air force would like to address ...	-1	Neutral
1	gregffff	chernihiv oblast ukrainians welcome their libe...	1	Positive
2	ThanapornThon17	america is preparing for something worse than...	0	Negative

```
text_leng = []
for text in df_new_en.text:
    tweet_leng = len(text.split())
    text_leng.append(tweet_leng)
```

```
df_new_en['text_leng'] = text_leng
```

```
df_new_en
```

	username	text	label	sentiment	text_leng
0	Yaniela	the ukrainian air force would like to address ...	-1	Neutral	29
1	gregffff	chernihiv oblast ukrainians welcome their libe	1	Positive	7

```
print(f" DF SHAPE: {df_new_en.shape}")
```

```
 DF SHAPE: (254626, 5)
```

```
df_new_en = df_new_en[df_new_en['text_leng'] > 4]
```

emails f...

```
tokenizer = BertTokenizerFast.from_pretrained('bert-base-uncased')
```

Downloading: 100% 28.0/28.0 [00:00<00:00, 465B/s]

Downloading: 100% 226k/226k [00:00<00:00, 2.33MB/s]

```
token_lens = []

for txt in df_new_en['text'].values:
    tokens = tokenizer.encode(txt, max_length=512, truncation=True)
    token_lens.append(len(tokens))

max_len=np.max(token_lens)
```

```
print(f"MAX TOKENIZED SENTENCE LENGTH: {max_len}")
```

```
MAX TOKENIZED SENTENCE LENGTH: 150
```

```
token_lens = []

for i,txt in enumerate(df_new_en['text'].values):
    tokens = tokenizer.encode(txt, max_length=512, truncation=True)
    token_lens.append(len(tokens))
    if len(tokens)>60:
        print(f"INDEX: {i}, TEXT: {txt}")
```

```
rrent price is 46162 indicators dailyrsi 620ma20 43160ma50 41464ma200 48293bollinger b
nian skif atgm in use against 2 tos1a thermobaric mrl destroying them both it seems th
nian skif atgm in use against 2 tos1a thermobaric mrl destroying them both it seems th
nian skif atgm in use against 2 tos1a thermobaric mrl destroying them both it seems th
rrent price is 46149 indicators dailyrsi 620ma20 43160ma50 41464ma200 48293bollinger b
```

```
nian skif atgm in use against 2 tos1a thermobaric mrl destroying them both it seems th
min eth gas tracking gweilow 66 gweiavg 67 gweihigh 67 gwei opensea salelow 4597avg 46
rrent price is 46150 indicators dailyrsi 620ma20 43160ma50 41464ma200 48293bollinger b
nian skif atgm in use against 2 tos1a thermobaric mrl destroying them both it seems th
eth contract address eft 0xb72962568345253f71a18318d67e13a282b187e6 contract address e
eth contract address eft 0xb72962568345253f71a18318d67e13a282b187e6 contract address e
eth contract address eft 0xb72962568345253f71a18318d67e13a282b187e6 contract address e
eth contract address eft 0xb72962568345253f71a18318d67e13a282b187e6 contract address e
eth contract address eft 0xb72962568345253f71a18318d67e13a282b187e6 contract address e
nian skif atgm in use against 2 tos1a thermobaric mrl destroying them both it seems th
nian skif atgm in use against 2 tos1a thermobaric mrl destroying them both it seems th
eth contract address eft 0xb72962568345253f71a18318d67e13a282b187e6 contract address e
eth contract address eft 0xb72962568345253f71a18318d67e13a282b187e6 contract address e
rrent price is 46161 indicators dailyrsi 620ma20 43161ma50 41464ma200 48293bollinger b
eth contract address eft 0xb72962568345253f71a18318d67e13a282b187e6 contract address e
eth contract address eft 0xb72962568345253f71a18318d67e13a282b187e6 contract address e
eth contract address eft 0xb72962568345253f71a18318d67e13a282b187e6 contract address e
eth contract address eft 0xb72962568345253f71a18318d67e13a282b187e6 contract address e
eth contract address eft 0xb72962568345253f71a18318d67e13a282b187e6 contract address e
eth contract address eft 0xb72962568345253f71a18318d67e13a282b187e6 contract address e
eth contract address eft 0xb72962568345253f71a18318d67e13a282b187e6 contract address e
eth contract address eft 0xb72962568345253f71a18318d67e13a282b187e6 contract address e
eth contract address eft 0xb72962568345253f71a18318d67e13a282b187e6 contract address e
eth contract address eft 0xb72962568345253f71a18318d67e13a282b187e6 contract address e
eth contract address eft 0xb72962568345253f71a18318d67e13a282b187e6 contract address e
eth contract address eft 0xb72962568345253f71a18318d67e13a282b187e6 contract address e
rrent price is 46168 indicators dailyrsi 620ma20 43160ma50 41464ma200 48293bollinger b
nian skif atgm in use against 2 tos1a thermobaric mrl destroying them both it seems th
nian skif atgm in use against 2 tos1a thermobaric mrl destroying them both it seems th
nian skif atgm in use against 2 tos1a thermobaric mrl destroying them both it seems th
nian skif atgm in use against 2 tos1a thermobaric mrl destroying them both it seems th
nian skif atgm in use against 2 tos1a thermobaric mrl destroying them both it seems th
sian army is visually confirmed to have lost three 220mm tos1a thermobaric mrls and tw
dded russian equipment losses 3x btr82a ifv 1x destroyed 2x captured1x 120mm 2s34 khos
sian troops continue a notsosuccessful retreat from oblast  the ukrainian forces destr
nian skif atgm in use against 2 tos1a thermobaric mrl destroying them both it seems th
nian skif atgm in use against 2 tos1a thermobaric mrl destroying them both it seems th
sian army is visually confirmed to have lost three 220mm tos1a thermobaric mrls and tw
nian skif atgm in use against 2 tos1a thermobaric mrl destroying them both it seems th
sian troops continue a notsosuccessful retreat from oblast  the ukrainian forces destr
nian skif atgm in use against 2 tos1a thermobaric mrl destroying them both it seems th
nian skif atgm in use against 2 tos1a thermobaric mrl destroying them both it seems th
sian troops continue a notsosuccessful retreat from oblast  the ukrainian forces destr
sian army is visually confirmed to have lost three 220mm tos1a thermobaric mrls and tw
nian skif atgm in use against 2 tos1a thermobaric mrl destroying them both it seems th

nian skif atgm in use against 2 tos1a thermobaric mrl destroying them both it seems th
sian troops continue a notsosuccessful retreat from oblast  the ukrainian forces destr
sian troops continue a notsosuccessful retreat from oblast  the ukrainian forces destr
dded russian equipment losses 3x btr82a ifv 1x destroyed 2x captured1x 120mm 2s34 khos
nian skif atgm in use against 2 tos1a thermobaric mrl destroying them both it seems th
nian skif atgm in use against 2 tos1a thermobaric mrl destroying them both it seems th
sian army is visually confirmed to have lost three 220mm tos1a thermobaric mrls and tw
nian skif atgm in use against 2 tos1a thermobaric mrl destroying them both it seems th
```

```
df_new_en['token_lens'] = token_lens
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
```

```
df_new_en = df_new_en.sort_values(by='token_lens', ascending=False)
df_new_en.head(20)
```

	username	text	label	sentiment	text_leng	to
26703	UnicefUkraine	please help ukrainian kids 1bfylqnwwwfacflts79...	-1	Neutral	9	
90393	crypto_tidings	stand with the people of now accepting cryptoc...	-1	Neutral	17	
46062	Ismailmurad17	we are a family from kharkov we were left with...	-1	Neutral	27	
93640	familyinukraine	we are a family from kharkov we were left wit...	-1	Neutral	27	
320962	CryptoNate54	eft efb eth contract address eft 0xb7296256834...	-1	Neutral	18	
321452	CryptoNate54	eft efb eth contract address eft 0xb7296256834...	-1	Neutral	18	
321424	CryptoNate54	eft efb eth contract address eft 0xb7296256834...	-1	Neutral	18	
321368	CryptoNate54	eft efb eth contract address eft 0xb7296256834...	-1	Neutral	18	
		eft efb eth contract				

```
df_new_en = df_new_en.iloc[:]
```

```
df_new_en.head()
```

	username	text	label	sentiment	text_leng	to
26703	UnicefUkraine	please help ukrainian kids 1bfylqnwwwfacflts79...	-1	Neutral	9	
90393	crypto_tidings	stand with the people of now accepting cryptoc...	-1	Neutral	17	

```
df_new_en = df_new_en.sample(frac=1).reset_index(drop=True)
```

```
df_new_en['label'].value_counts()
```

```
-1    110772
 0     78585
 1     53843
Name: label, dtype: int64
```

```
df_Positive=df_new_en[df_new_en['label']==1]
df_Neutral=df_new_en[df_new_en['label']==-1]
df_Negative=df_new_en[df_new_en['label']==0]
```

```
df_Negative_downsampled=df_Negative.sample(df_Positive.shape[0])
df_Neutral_downsampled=df_Neutral.sample(df_Positive.shape[0])
```

```
df_balanced = pd.concat([df_Negative_downsampled, df_Neutral_downsampled, df_Positive])
```

```
df_balanced.head()
```

	username	text	label	sentiment	text_leng	token_lens
35758	MurielVieux	on the ground russias invasion of ukraine phot...	0	Negative	34	37
		russians				

Plotting the word cloud of tweets labelled as positive

```
df_positive=df_balanced[df_balanced['label'] == 1]

df_positive.head()
```

	username	text	label	sentiment	text_leng	token_lens
2	PalmaOksana	in this video soldiers were filming a videose...	1	Positive	17	22
3	Ewe_Paz_HeT	twitter is a way of being a dictator for	1	Positive	21	28

```
df_positive['text'].to_csv('pos_tweets.csv',index=False)

stylecloud.gen_stylecloud(file_path='pos_tweets.csv',
                          icon_name='fab fa-twitter',
                          palette='colorbrewer.qualitative.Paired_3',
                          background_color='white',
                          gradient='vertical',
                          stopwords=True,
                          custom_stopwords=['philipvollet','NFT','MINT','RT','THE','IS','WITH
                         )

from IPython.display import Image
Image('stylecloud.png')
```

Plotting the word cloud of tweets labelled as negative

```
df_negative=df_balanced[df_balanced['label'] == 0]

df_negative.head()
```

	username	text	label	sentiment	text_leng	token_lens
35758	MurielVieux	on the ground russias invasion of ukraine phot...	0	Negative	34	37
233772	jrmichaluk	russians have reportedly left antonov airport ...	0	Negative	24	27
85184	melindaharing	team in said in last few minutes been given a...	0	Negative	42	48
[illegible]	[illegible]	early today 2 ukrainian mi24	[illegible]	[illegible]	[illegible]	[illegible]

```
df_negative['text'].to_csv('neg_tweets.csv',index=False)

stylecloud.gen_stylecloud(file_path='neg_tweets.csv',
                          icon_name='fab fa-twitter',
                          palette='colorbrewer.qualitative.Paired_3',
                          background_color='white',
                          gradient='vertical',
                          stopwords=True,
                          custom_stopwords=['philipvollet','NFT','TASTE','MINT','RT','THE','I
                         )

from IPython.display import Image
Image('stylecloud.png')
```

Building the text classifier model

```
X=df_balanced['text'].values
y=df_balanced['label'].values
```

```
X_train, X_valid, y_train, y_valid = train_test_split(X,y,test_size=0.1,stratify=y, random_st
```

```
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.2, random_state=42)
```

```
y_train_le=y_train.copy()
y_valid_le=y_valid.copy()
y_test_le=y_test.copy()
```

```
ohe=preprocessing.OneHotEncoder()
y_train=ohe.fit_transform(np.array(y_train).reshape(-1,1)).toarray()
y_valid=ohe.fit_transform(np.array(y_valid).reshape(-1,1)).toarray()
y_test=ohe.fit_transform(np.array(y_test).reshape(-1,1)).toarray()
```

```
print(f"TRAINING DATA:{X_train.shape[0]}\nVALIDATION DATA: {X_valid.shape[0]}\nTESTING DATA:
```

```
TRAINING DATA:129223
VALIDATION DATA: 16153
TESTING DATA: 32306
```

```
clf = CountVectorizer()
X_train_cv =  clf.fit_transform(X_train)
X_test_cv = clf.transform(X_test)
```

```
tf_transformer = TfidfTransformer(use_idf=True).fit(X_train_cv)
X_train_tf = tf_transformer.transform(X_train_cv)
X_test_tf = tf_transformer.transform(X_test_cv)
```

```
nb_clf = MultinomialNB()
```

```
nb_clf.fit(X_train_tf, y_train_le)
```

```
MultinomialNB()
```

```
nb_pred=nb_clf.predict(X_test_tf)
```

```
print('\tClassification Report for Naive Bayes:\n\n',(classification_report(y_test_le,nb_pred
```

```
            Classification Report for Naive Bayes:

                   precision    recall  f1-score   support

              -1       0.94      0.78      0.85     10725
               0       0.85      0.91      0.88     10634
               1       0.81      0.90      0.86     10947

        accuracy                           0.86     32306
       macro avg       0.87      0.86      0.86     32306
    weighted avg       0.87      0.86      0.86     32306
```

Printing the confusion matrix for Naive Bayes Classifier

```
print('Confusion matrix\n',confusion_matrix(y_test_le,nb_pred))
```

```
Confusion matrix
 [[8340  987 1398]
 [ 157 9634  843]
 [ 379  706 9862]]
```

Our base Naive Bayes model gives us an accuracy of 86%

Now, we'll build a pipeline and try out different models to find out the best classification model for our data

```
sample_size = int(len(df_balanced)*0.05)
sampleDf = df_balanced.sample(sample_size, random_state=23)
X = sampleDf.text.values
```

```
y = sampleDf.label.values

# X=df_balanced.text
# y=df_balanced.label


X_train, X_test, y_train, y_test=train_test_split(X, y, test_size=0.20, random_state=42)


print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(6460,)
(1616,)
(6460,)
(1616,)
```

```
from sklearn.pipeline import make_pipeline
from sklearn.naive_bayes import MultinomialNB,ComplementNB
from sklearn.linear_model import LogisticRegression, RidgeClassifier
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.tree import DecisionTreeClassifier

pipelines=[]
for model in [DecisionTreeClassifier(), MultinomialNB(), ComplementNB(),
              LogisticRegression(solver='saga'), RidgeClassifier(solver='auto'), SVC(),Random
    pipeline=make_pipeline(TfidfVectorizer(), model)
    pipelines.append(pipeline)


#Training the model
import time
training_time=[]
for pipeline in pipelines:
    start=time.time()
    pipeline.fit(X_train, y_train)
    stop=time.time()
    training_time.append(stop-start)


#Prediction from test dataset
from sklearn.metrics import classification_report, confusion_matrix, f1_score, precision_scor
model_name=[]
precision_array=[]
recall_array=[]
f1_array=[]
test_time=[]
print("Classifiation Report\n")
```

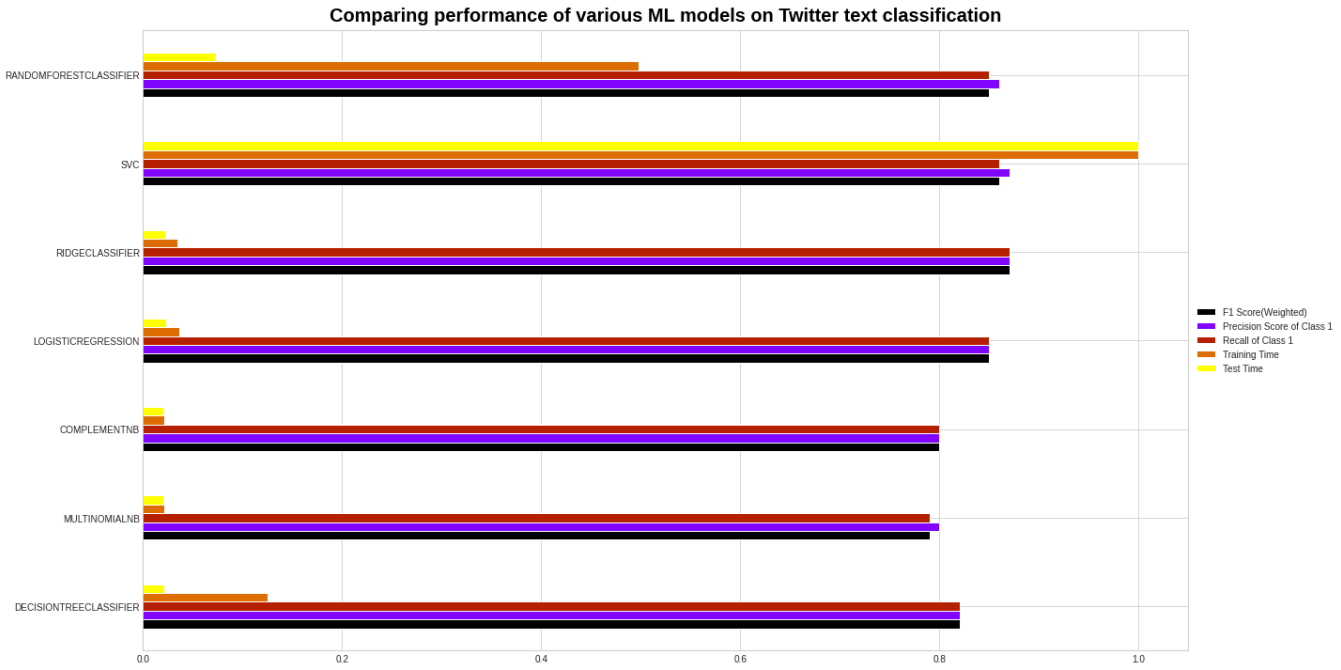
```
print("*****************************************************")
for i, pipeline in enumerate(pipelines):
    start=time.time()
    y_pred=pipeline.predict(X_test)
    stop=time.time()
    test_time.append(stop-start)
    print(pipelines[i].steps[1][0].upper())
    model_name.append(pipelines[i].steps[1][0].upper())
    f1_array.append(round(f1_score(y_test, y_pred, average='weighted'),2))
    precision_array.append(round(precision_score(y_test, y_pred, average='weighted'),2))
    recall_array.append(round(recall_score(y_test, y_pred, average='weighted'),2))
    print("\n",classification_report(y_test, y_pred))
    print("*****************************************************")
```

```
 Classifiation Report

 ****************************************************
 DECISIONTREECLASSIFIER

               precision    recall  f1-score   support

          -1       0.80      0.80      0.80       508
           0       0.87      0.87      0.87       564
           1       0.79      0.80      0.80       544

    accuracy                           0.82      1616
   macro avg       0.82      0.82      0.82      1616
weighted avg       0.82      0.82      0.82      1616

 ****************************************************
 MULTINOMIALNB

               precision    recall  f1-score   support

          -1       0.86      0.67      0.75       508
           0       0.77      0.85      0.81       564
           1       0.76      0.85      0.80       544

    accuracy                           0.79      1616
   macro avg       0.80      0.79      0.79      1616
weighted avg       0.80      0.79      0.79      1616

 ****************************************************
 COMPLEMENTNB

               precision    recall  f1-score   support

          -1       0.83      0.70      0.76       508
           0       0.80      0.86      0.83       564
           1       0.78      0.83      0.80       544

    accuracy                           0.80      1616
   macro avg       0.80      0.79      0.80      1616
weighted avg       0.80      0.80      0.80      1616
```

```
**************************************************
LOGISTICREGRESSION

              precision    recall  f1-score   support

          -1       0.83      0.81      0.82       508
           0       0.92      0.88      0.90       564
           1       0.81      0.86      0.83       544

    accuracy                           0.85      1616
   macro avg       0.85      0.85      0.85      1616
weighted avg       0.85      0.85      0.85      1616

**************************************************
RIDGECLASSIFIER

              precision    recall  f1-score   support
```

Plotting the performance

```
#Plotting the various performance metrix of all models
training_time=np.array(training_time)/np.max(training_time)
test_time=np.array(test_time)/np.max(test_time)
score_df=pd.DataFrame({'F1 Score(Weighted)':f1_array,
                    'Precision Score of Class 1':precision_array,
                    'Recall of Class 1':recall_array,
                     'Training Time': training_time,
                     'Test Time':test_time}, index=model_name)

f=plt.figure(figsize=(20,10))
plt.title('Comparing performance of various ML models on Twitter text classification', color=
score_df.plot(kind='barh', ax=f.gca(), cmap='gnuplot')

plt.legend(loc='center left', bbox_to_anchor=(1.0, 0.5))
plt.show()
```



Conclusion

In our approach to analyze twitter data sentiment of Russia-Ukraine invasion, we under took the following approach:

After the data processing and labeling of the dataset we found that:

1. The dataset had a majority of neutral sentiment, followed by negative and then positive.
2. As most tweets have a sentiment of either neutral or negative, we can infer that the majority is not in favour of the invasion.
3. As twitter data includes users from all over the world who are not directly affected by the invasion, as a result high number of tweets fall under the neutral category.
4. For classification we have built our base model on Naive Bayes which resulted in an accuracy of 86%
5. Following this, to test our data on all models, we developed a pipeline and trained our model on 6 classification models. After applying the pipeline, we found that Ridge Classifier out-performed all other models with an accuracy of 89%

Future Work

For future analysis, we use transformers to carry out sentiment analysis. We can make use of BERT and RoBERTa models for obtaining higher accuracy.