Fr. Conceicao Rodrigues College of Engineering

Father Agnel Ashram, Bandstand, Bandra –west, Mumbai-50

Department of Computer Engineering

SOCIAL MEDIA ANALYTICS LAB

Experiment No: 3

Aim: Data Cleaning and Storage- Preprocess, filter and store social media data for business

Objective: To perform preprocessing on social media data and make it ready for analysis.

Lab outcomes:

At the end of this lab session, students will be able to…

1. Clean and preprocess the data captured from social media.
2. Perform the exploratory data analysis.

Theory:

● Data cleaning and preprocessing is an essential – and often crucial – part of any analytical process. Social media contains different types of data: information about user profiles, statistics

● (number of likes or number of followers), verbatims, and other media content.

● Quantitative data is very convenient for an analysis using statistical and numerical methods, but unstructured data such as user comments is much more challenging.

● To get meaningful information, one has to perform the whole process of information retrieval. It starts with the definition of the data type and data structure.

● On social media, unstructured data is related to text, images, videos, and sound and we will mostly deal with textual data.

● Then, the data has to be cleaned and normalized.

Preprocessing

● Preprocessing is one of the most important parts of the analysis process.

● It reformats the unstructured data into uniform, standardized form.

● The characters, words, and sentences identified at this stage are the fundamental units passed to all further processing stages.

● The quality of the preprocessing has a big impact of the final result on the whole process.

SOCIAL MEDIA ANALYTICS LAB

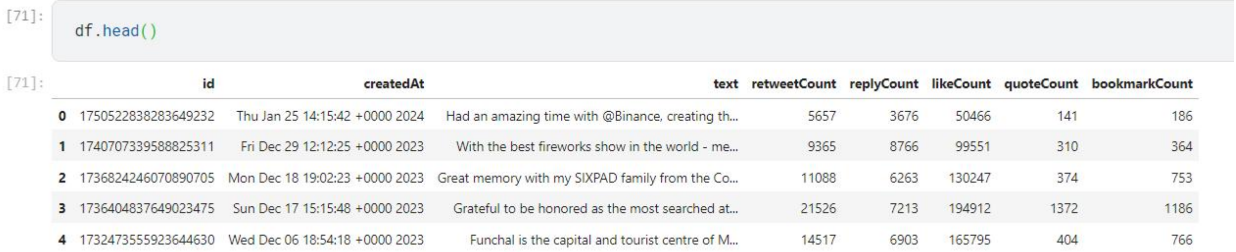
● There are several stages of the process: from simple text cleaning by removing white spaces, punctuation, HTML tags and special characters up to more sophisticated normalization techniques such as tokenization, stemming or lemmatization.

Steps :

Step 1: Loading Packages
Step 2: X's Data extraction using snscrape Python library
Step 3: X's Data Cleaning and Preprocessing using Python
Step 4: X's Data Visualization
Step 5: Twitter Data Sentiment Analysis using Textblob.

Student's Tasks

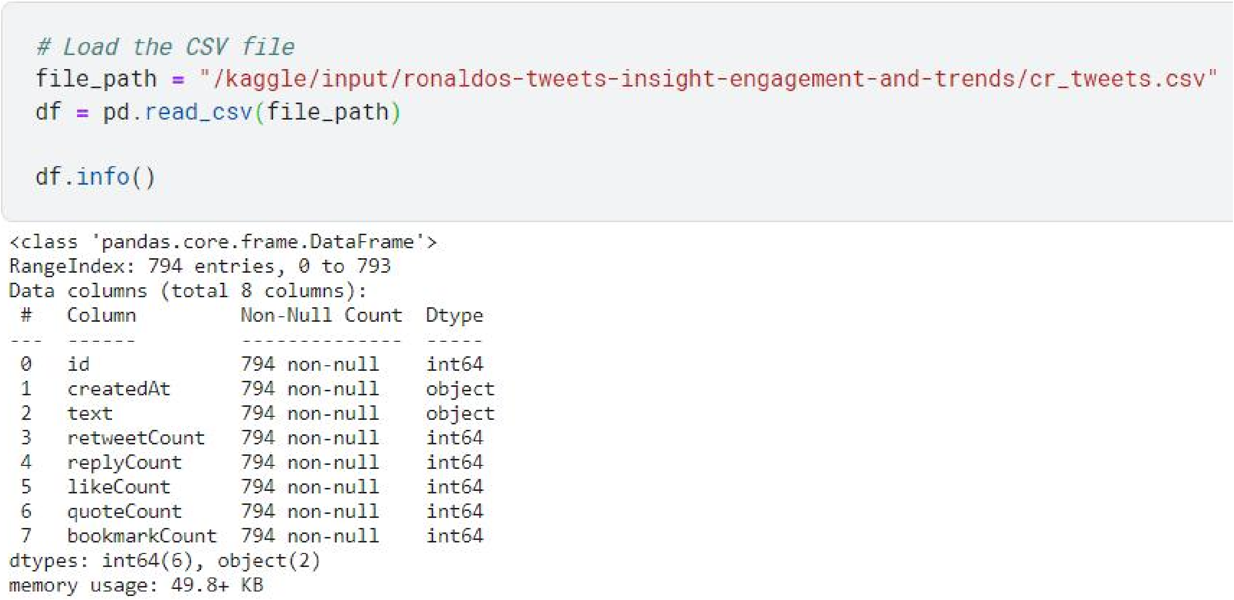
1. Scrape X's Data for Ronaldos tweets insight engagement and trends 2024
2. Create document corpus with tweet text



```
[71]: df.head()
```

[71]:

	id	createdAt	text	retweetCount	replyCount	likeCount	quoteCount	bookmarkCount
0	1750522838283649232	Thu Jan 25 14:15:42 +0000 2024	Had an amazing time with @Binance, creating th...	5657	3676	50466	141	186
1	1740707339588825311	Fri Dec 29 12:12:25 +0000 2023	With the best fireworks show in the world - me...	9365	8766	99551	310	364
2	1736824246070890705	Mon Dec 18 19:02:23 +0000 2023	Great memory with my SIXPAD family from the Co...	11088	6263	130247	374	753
3	1736404837649023475	Sun Dec 17 15:15:48 +0000 2023	Grateful to be honored as the most searched at...	21526	7213	194912	1372	1186
4	1732473555923644630	Wed Dec 06 18:54:18 +0000 2023	Funchal is the capital and tourist centre of M...	14517	6903	165795	404	766



```
# Load the CSV file
file_path = "/kaggle/input/ronaldos-tweets-insight-engagement-and-trends/cr_tweets.csv"
df = pd.read_csv(file_path)

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 794 entries, 0 to 793
Data columns (total 8 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   id             794 non-null    int64 
 1   createdAt      794 non-null    object
 2   text           794 non-null    object
 3   retweetCount   794 non-null    int64 
 4   replyCount     794 non-null    int64 
 5   likeCount      794 non-null    int64 
 6   quoteCount     794 non-null    int64 
 7   bookmarkCount  794 non-null    int64 
dtypes: int64(6), object(2)
memory usage: 49.8+ KB
```

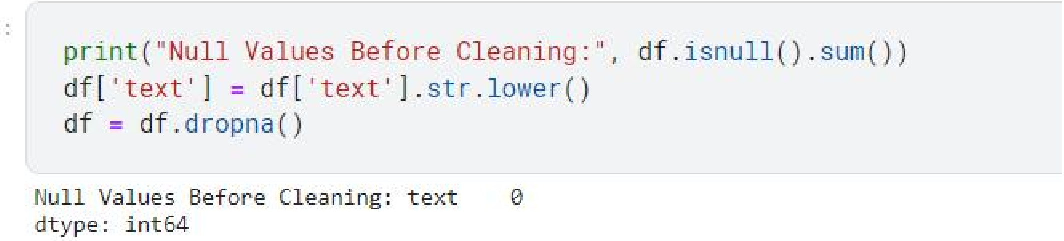
3. Data Cleaning & Preprocessing-

    3.1 Convert text to Lower Case

    3.2 Remove the links (URLs)

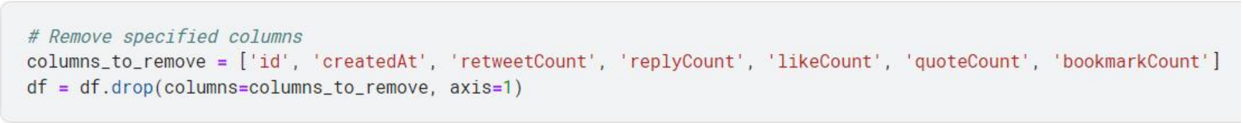
    3.3 Remove anything except the English language and space.

    3.4 Remove Stop words.

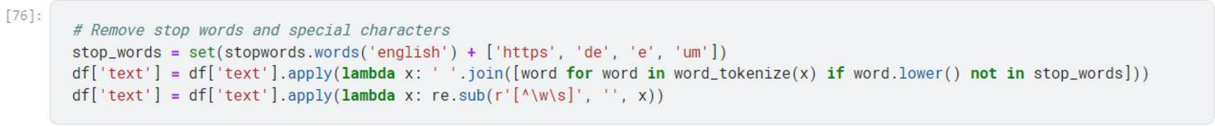


```
print("Null Values Before Cleaning:", df.isnull().sum())
df['text'] = df['text'].str.lower()
df = df.dropna()
```

```
Null Values Before Cleaning: text    0
dtype: int64
```

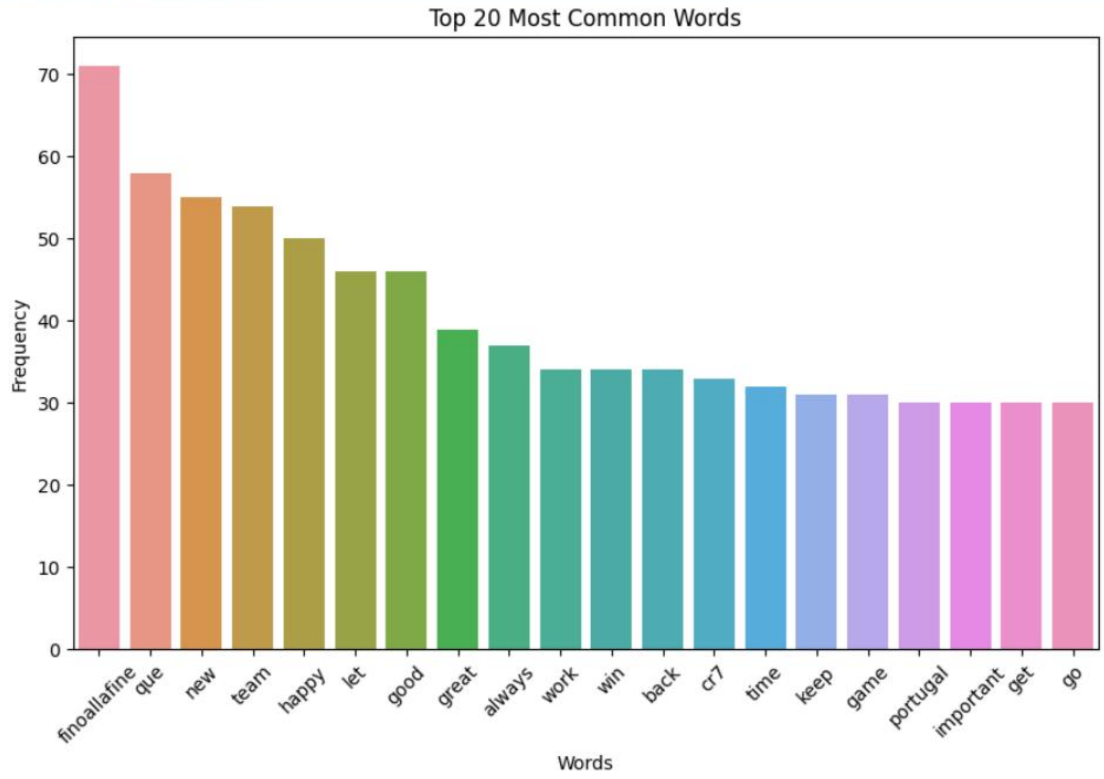


```
# Remove specified columns
columns_to_remove = ['id', 'createdAt', 'retweetCount', 'replyCount', 'likeCount', 'quoteCount', 'bookmarkCount']
df = df.drop(columns=columns_to_remove, axis=1)
```

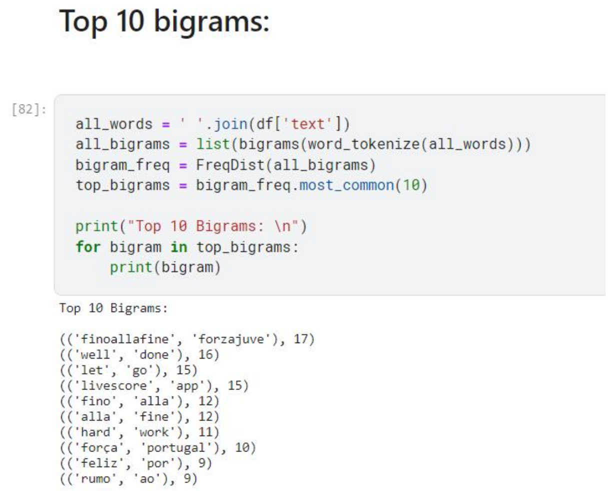


```
[76]:
# Remove stop words and special characters
stop_words = set(stopwords.words('english') + ['https', 'de', 'e', 'um'])
df['text'] = df['text'].apply(lambda x: ' '.join([word for word in word_tokenize(x) if word.lower() not in stop_words]))
df['text'] = df['text'].apply(lambda x: re.sub(r'[^\w\s]', '', x))
```

4. Visualize top 20 most common words.



5. Visualize top 10 bigrams.

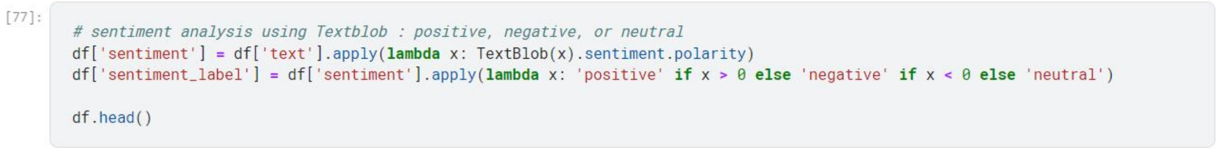


Top 10 bigrams:

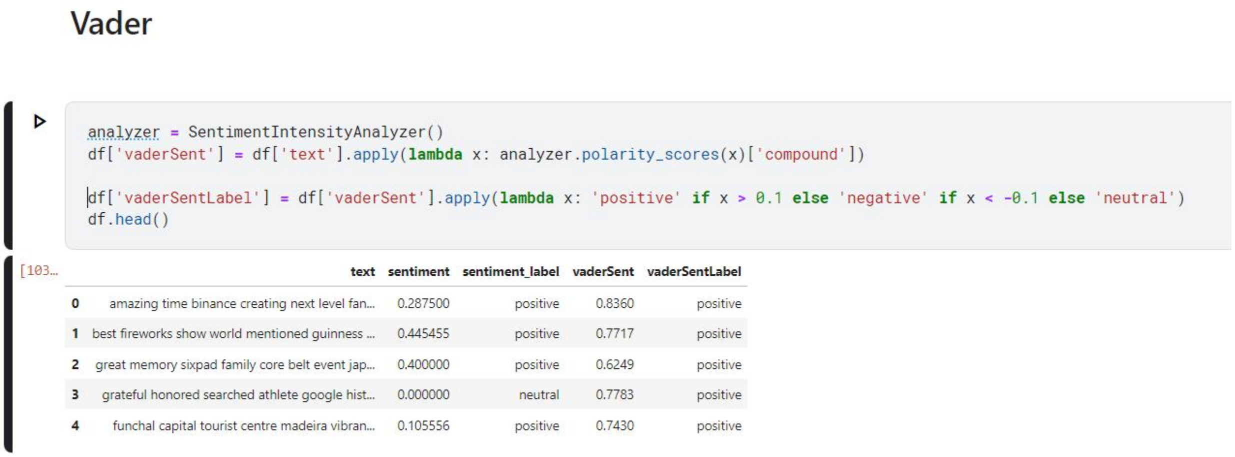
```
[82]:
all_words = ' '.join(df['text'])
all_bigrams = list(bigrams(word_tokenize(all_words)))
bigram_freq = FreqDist(all_bigrams)
top_bigrams = bigram_freq.most_common(10)

print("Top 10 Bigrams: \n")
for bigram in top_bigrams:
    print(bigram)
```

```
Top 10 Bigrams:

(('finoallafine', 'forzajuve'), 17)
(('well', 'done'), 16)
(('let', 'go'), 15)
(('livescore', 'app'), 15)
(('fino', 'alla'), 12)
(('alla', 'fine'), 12)
(('hard', 'work'), 11)
(('força', 'portugal'), 10)
(('feliz', 'por'), 9)
(('rumo', 'ao'), 9)
```

6. Perform the sentiment analysis using textblob.



```
[77]:
# sentiment analysis using Textblob : positive, negative, or neutral
df['sentiment'] = df['text'].apply(lambda x: TextBlob(x).sentiment.polarity)
df['sentiment_label'] = df['sentiment'].apply(lambda x: 'positive' if x > 0 else 'negative' if x < 0 else 'neutral')

df.head()
```

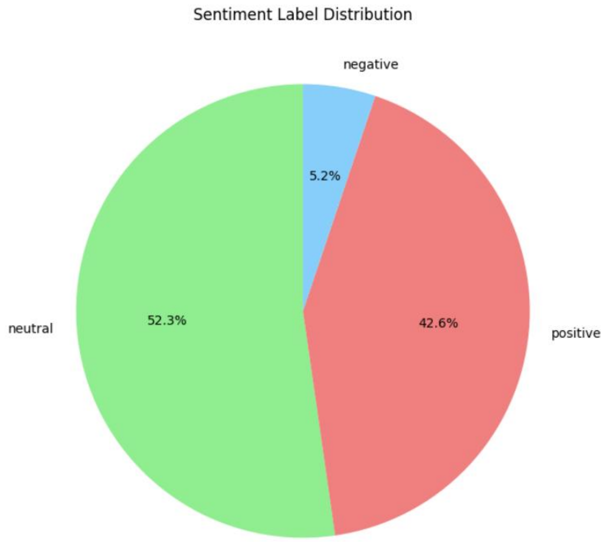


Vader

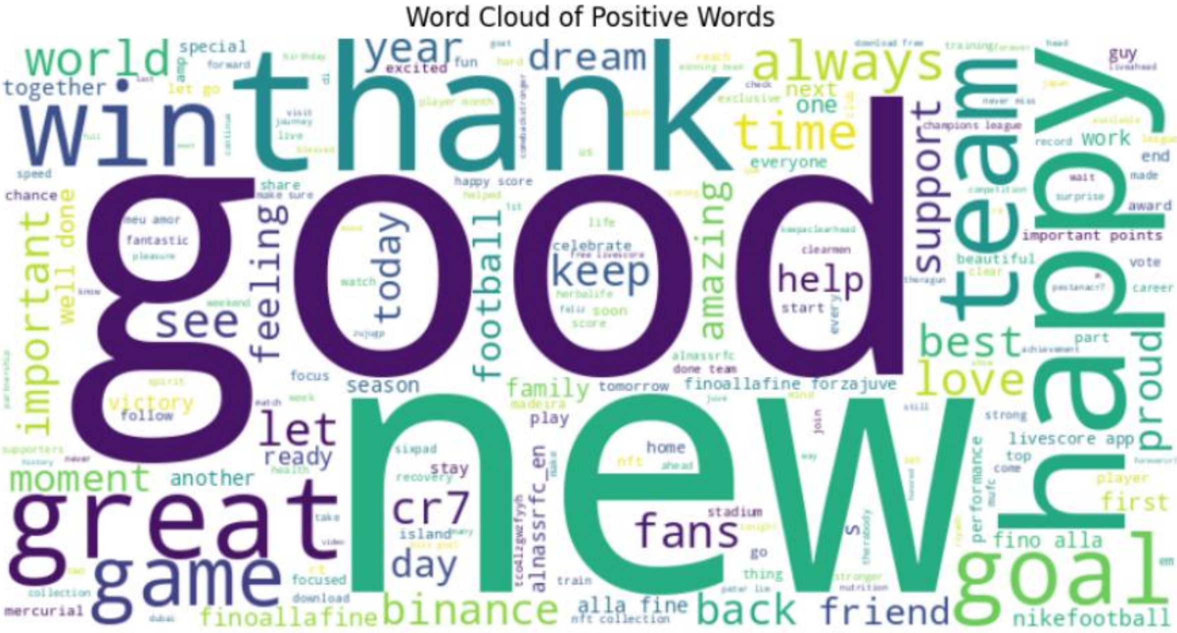
```
analyzer = SentimentIntensityAnalyzer()
df['vaderSent'] = df['text'].apply(lambda x: analyzer.polarity_scores(x)['compound'])

df['vaderSentLabel'] = df['vaderSent'].apply(lambda x: 'positive' if x > 0.1 else 'negative' if x < -0.1 else 'neutral')
df.head()
```

[103...

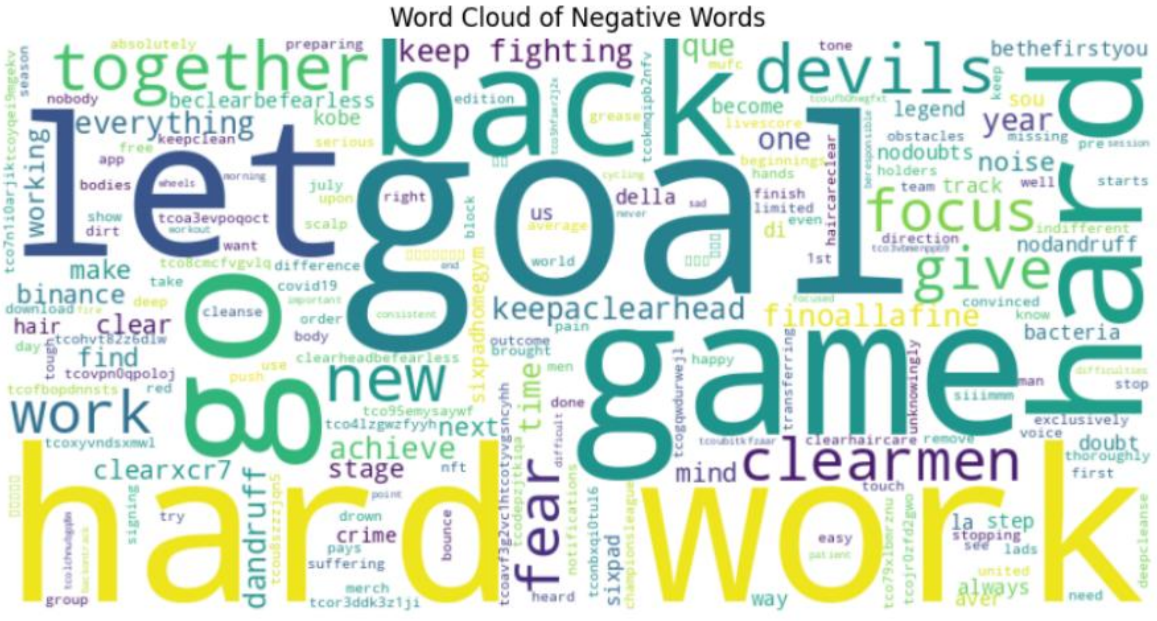
	text	sentiment	sentiment_label	vaderSent	vaderSentLabel
0	amazing time binance creating next level fan...	0.287500	positive	0.8360	positive
1	best fireworks show world mentioned guinness ...	0.445455	positive	0.7717	positive
2	great memory sixpad family core belt event jap...	0.400000	positive	0.6249	positive
3	grateful honored searched athlete google hist...	0.000000	neutral	0.7783	positive
4	funchal capital tourist centre madeira vibran...	0.105556	positive	0.7430	positive


Sentiment Label Distribution
negative
5.2%
neutral
52.3%
42.6%
positive

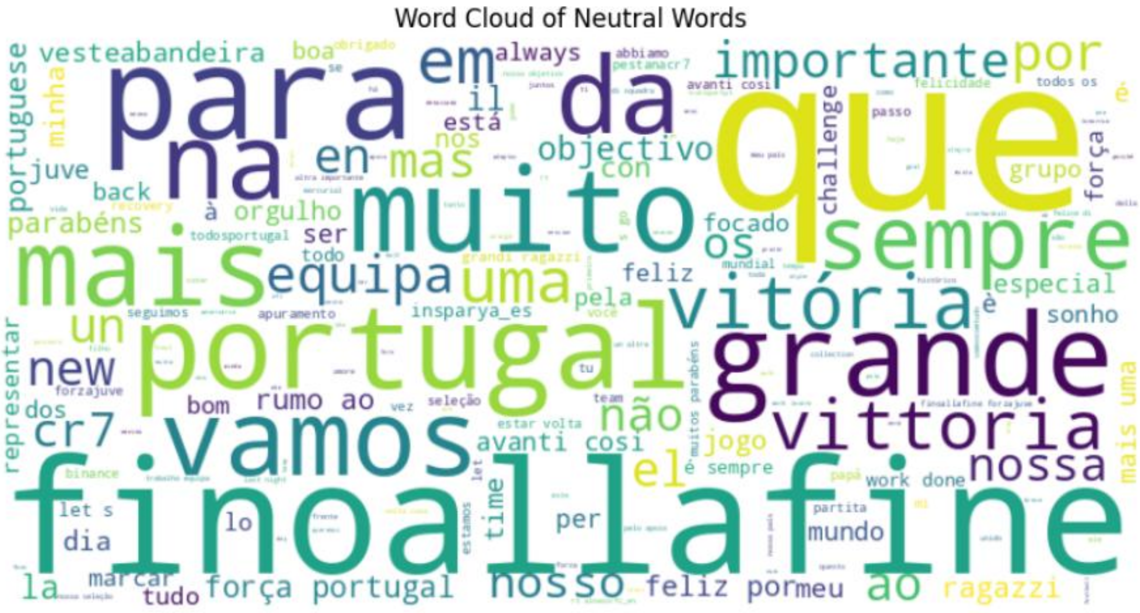
7. Display the word cloud of positive words.



8. Display the word cloud of negative words.



9. Display the word cloud of neutral words.



Word Cloud of Neutral Words

Kaggle Link (for Code) :

https://www.kaggle.com/mrappplg/sma-exp-3-v1