SENTIMENTAL ANALYSIS

Presented by

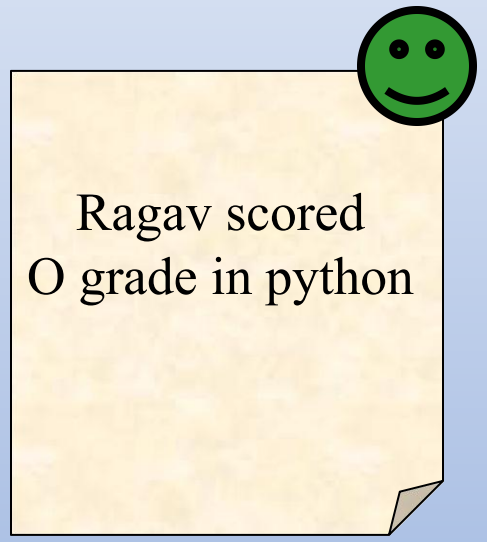
A.venkatesh(21pa1a0502)
G.Charan Sai(21pa1a0547)
A.V.Ravindra(21pa1a0506)

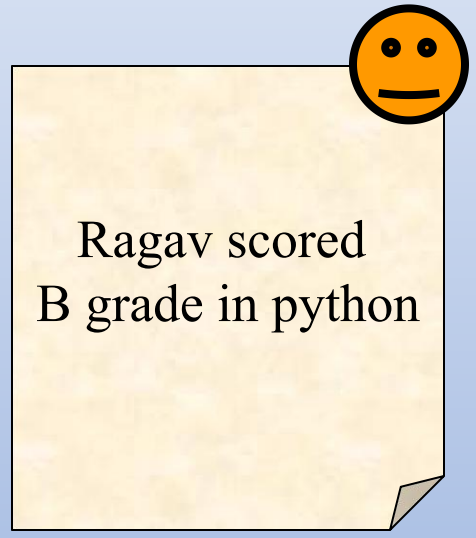
College name

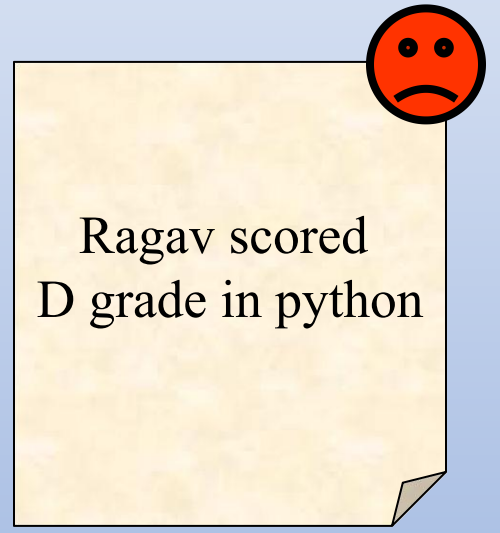
Vishnu institute of technology

What is Sentimental Analysis?

. Identify the orientation of opinion in a piece of text







. Can be generalized to a wider set of emotions

Motivation

. Knowing sentiment is a very natural ability of a human being.

Can a machine be trained to do it?

. SA aims at getting sentiment-related knowledge especially from the huge amount of information on the internet

. Can be generally used to understand opinion in a set of documents

Theme

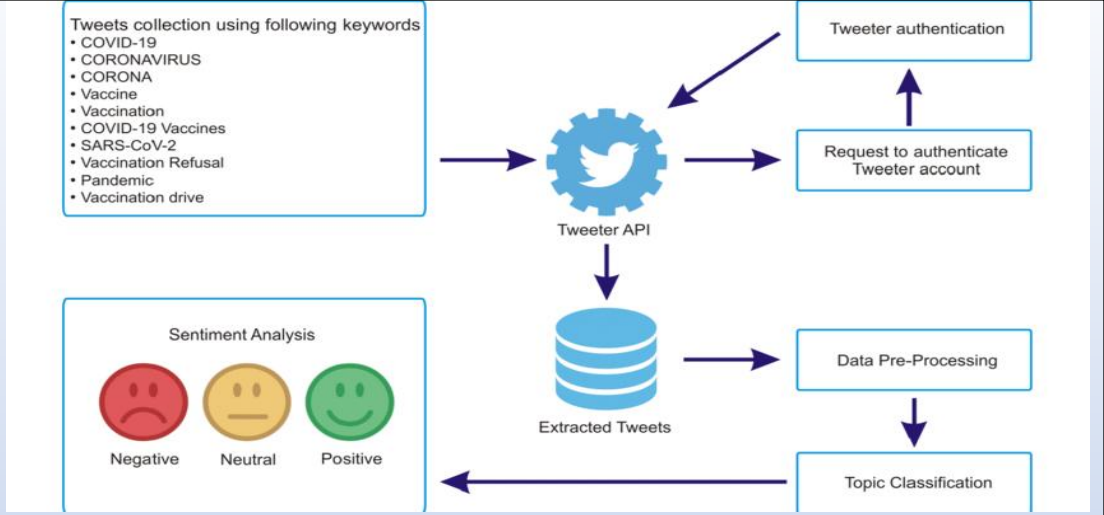
COVID-19 BOOSTER VACCINATION SHOTS

Why Covid-19 Booster Vaccination Shots ?

The COVID-19 outbreak a pandemic, which poses a serious threat to global public health and result in a tsunami of online social media. Individuals frequently express their views, opinions and emotions about the events of the pandemic on Twitter, Facebook, etc. Many researches try to analyze the sentiment of the COVID-19-related content from these social networks. However, they have rarely focused on the vaccine. In this ppt, we study the COVID-19 vaccine topic from Twitter.

Platforms used for extracting the data?

.TWITTER



Tags used to extract the data ?

#COVID19Vaccine
#antivaxxers
#COVID19
#vaccine
#vaccineinjuries
#VaccineDeaths
#VaccineSideEffects
#vaccination
#pandemic

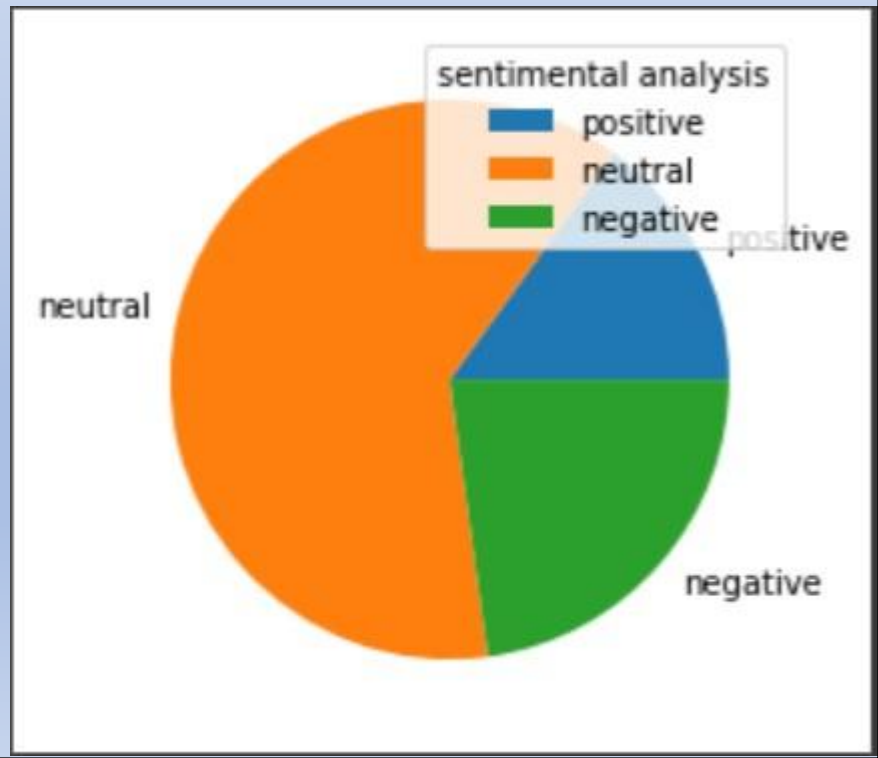
Why Twitter Data for Sentiment Analysis?

- Popular microblogging site
- Short Text Messages of 140 characters
- 240+ million active users
- 500 million tweets are generated everyday
- Twitter audience varies from common man to celebrities
- Users often discuss current affairs and share personal views on various subjects
- Tweets are small in length and hence unambiguous

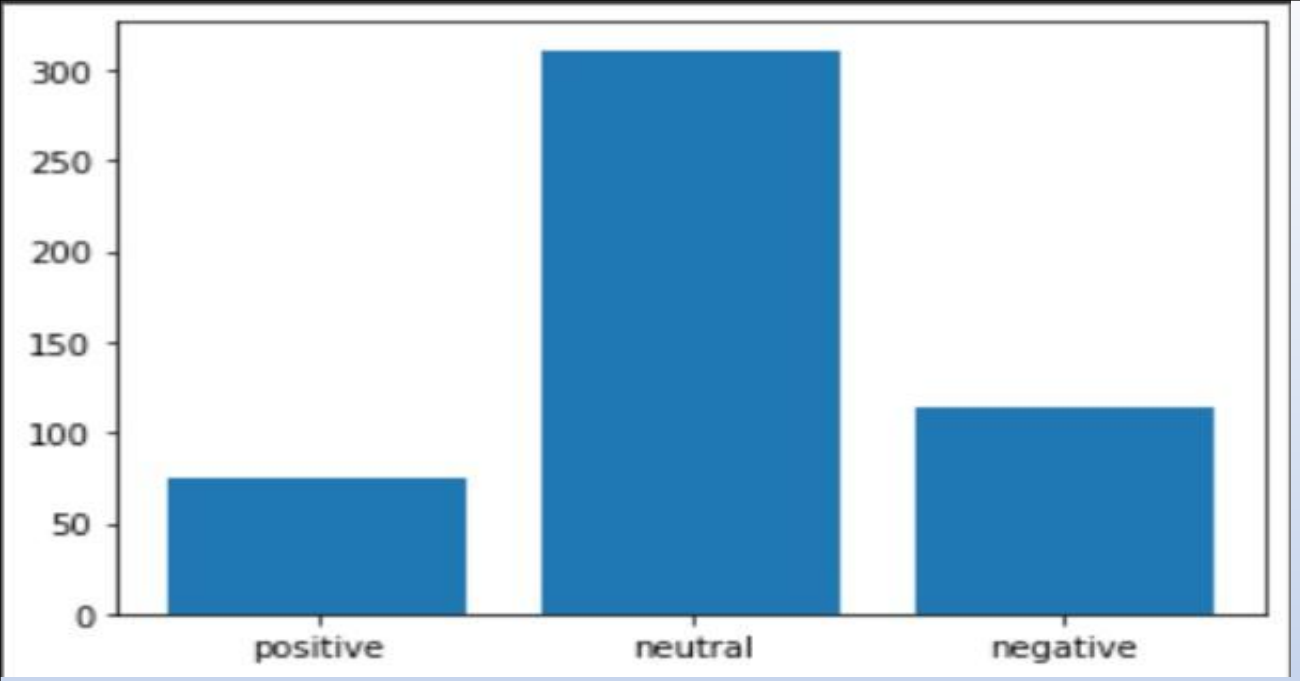
APIs or libraries used to perform data extraction?

.numpy
.math
.pandas
.snscrape
.transformers
.scipy.special
.re

Statistics of the extracted data.

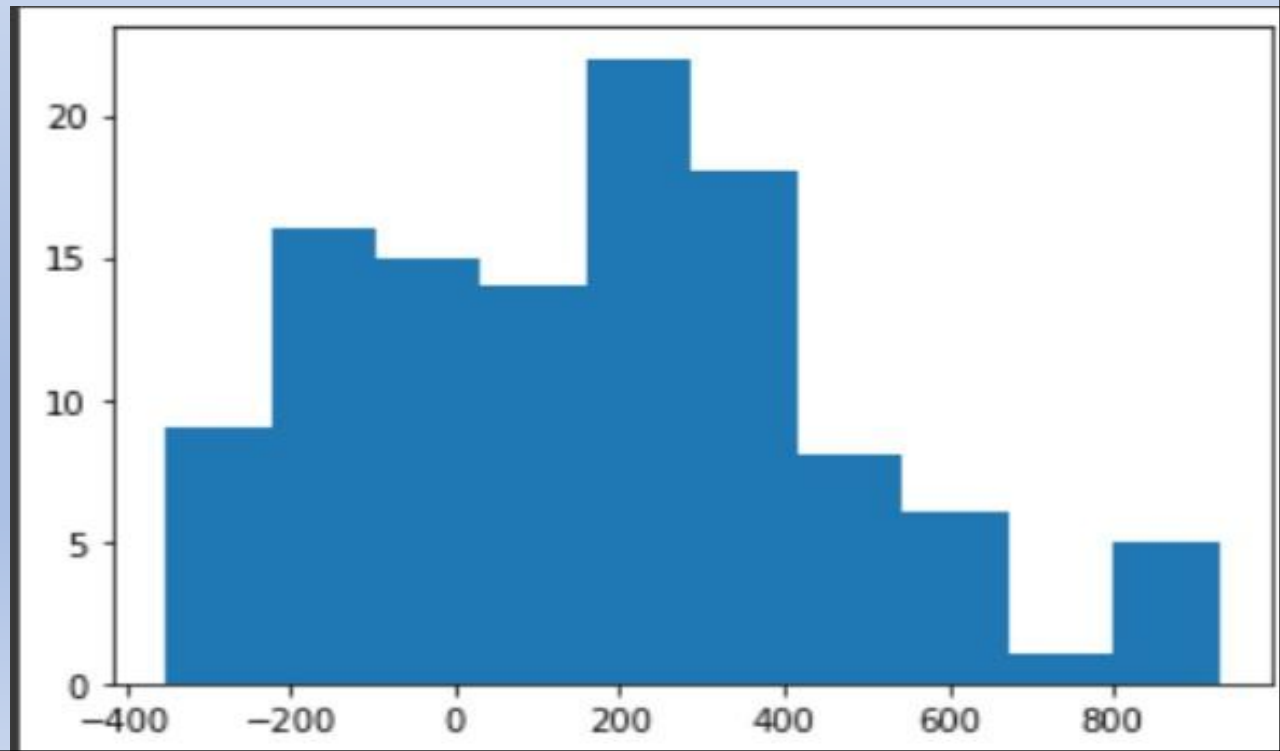


Sentimental analysis pie chart on covid-19



Sentimental analysis bar graph on covid-19

Sentimental analysis histograms on covid-19



Is the collected data clean or noisy?

THE COLLECTED DATA IS NOISY AND CONTAIN DIFFERENT LANGUAGES ,SPECIAL CHARACTERS AND EMOJIS. BY USING RESPECTIVE LIBRARIES TO MAKE THE DATA CLEAN AND EFFICIENT.

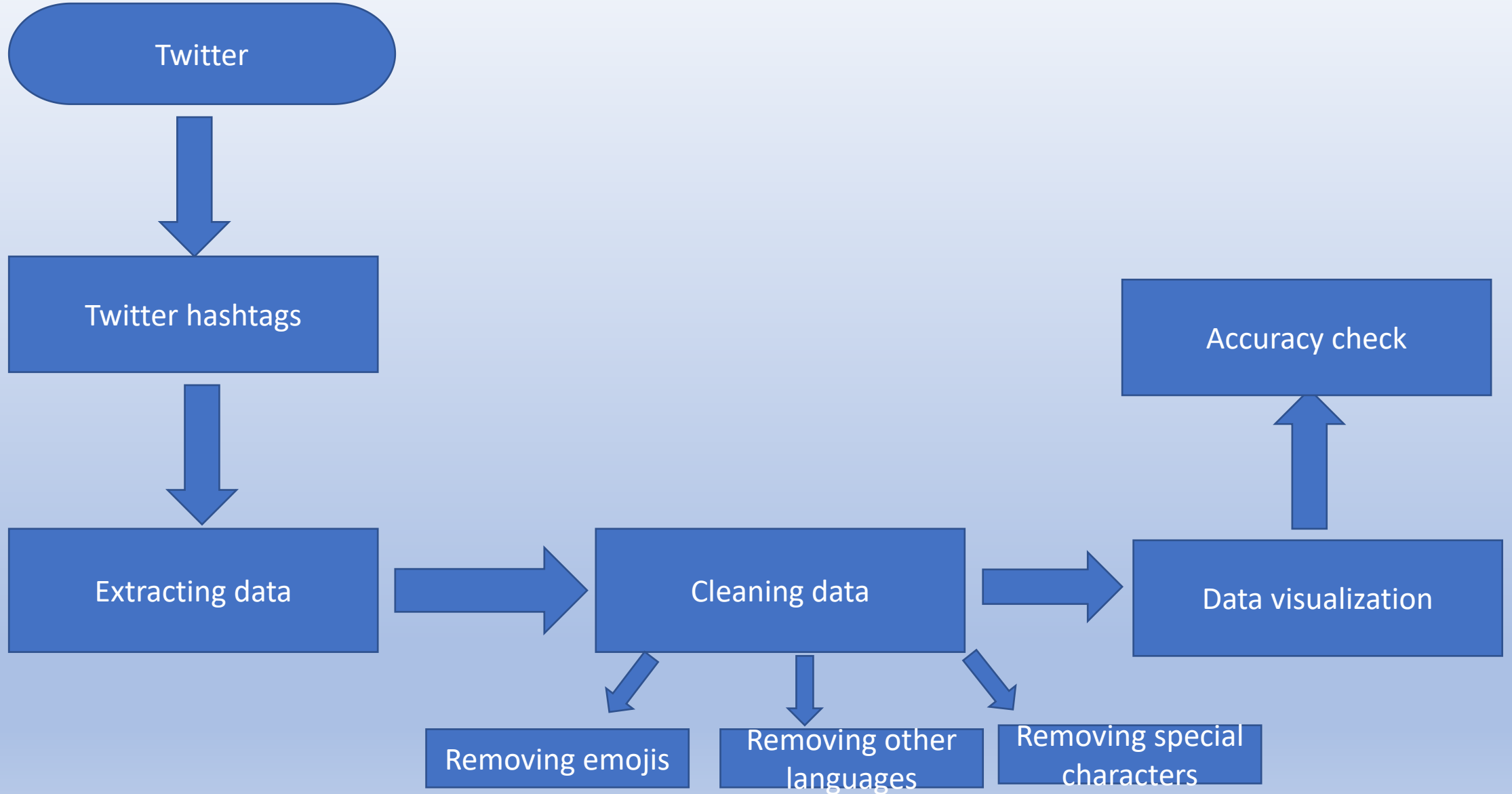
DATA CLEANING

What are the data cleaning strategies you have followed and why?

To clean the data we imported re library and to delete the extra symbols and emojis we used uniary code

```
import re

text = u'This dog \U0001f602'
x='😂 is  😊 the jhkjndjh'


emoji_pattern = re.compile("["
        u"\U0001F600-\U0001F64F"  # emoticons
        u"\U0001F300-\U0001F5FF"  # symbols & pictographs
        u"\U0001F680-\U0001F6FF"  # transport & map symbols
        u"\U0001F1E0-\U0001F1FF"  # flags (iOS)
        u"\U0001F600-\U0001F64F"
        u"\U00000531-\U00000DF4"
        u"\U00000900-\U0000097F"
        u"\U00000021-\U0000002F"
        u"\U0000003A-\U00000040"
        u"\U0000007B-\U000000FF"
        u"\U0000005B-\U00000060"
        u"\U00000021"
        u"\U0000003F"
        u"\U0000005F"
        u"\U00000022"
        u"\U0000003A"
        u"\U00000027"
        u"\U0000003B"
        u"\U0001F974"
        u"\U0001FA94"
        u"\U00002639"
        u"\U00002764"
        u"\U0001F9D0"
        u"\U0001F9A0"
        u"\U00002620"
        u"\U0001F92C"
        u"\U0001F923"
        u"\U0001F975"
        u"\U0001F91E"
```

```
        u"\U00002714"
        u"\U0001F9EA"
        u"\U0001F929"
        u"\U0000270A"
        u"\U00002744"
        u"\U00002705"
        u"\U00002B55"
        u"\U0001F914"
        u"\U00002640"
        u"\U00002611"
        u"\U0000260B"
        u"\U0001F911"
        u"\U0001F928"
        u"\U0001F914"
        u"\U0001F92B"
        u"\U0001F95F"
        u"\U00002795"
        u"\U00002642"
        u"\U00002708"
        u"\U0001F92A"
        u"\U00002B50"
        u"\U0001F937"
        u"\U00002019"
        u"\U0000270B"
        u"\U0000201D"
        u"\U0000201C"
        u"\U0001FAA7"
        u"\U0001F7CA"
        u"\U00002605"
        u"\U00002614"
        u"\U000026AA"



                          "]+", flags=re.UNICODE)
print(emoji_pattern.sub(r'', text)) # no emoji
print(data[1])
print(emoji_pattern.sub(r'',data[1]))
print(emoji_pattern.sub(r'',x))
for i in range(len(data)):
  data[i]=emoji_pattern.sub(r'',data[i])
  print(data[i])
```


Twitter
Twitter hashtags
Extracting data
Cleaning data
Removing emojis
Removing other languages
Removing special characters
Data visualization
Accuracy check

Sentiment Analysis

EXPLAIN THE TECHNIQUE OR MODELLING APPROACH YOU HAVE FOLLOWED TO ARRIVE AT A SENTIMENT OF THE POST (POSITIVE, NEUTRAL, OR NEGATIVE). STATE THE REASON FOR FOLLOWING THAT TECHNIQUE ?

```
install transformers and scipy.special .later
import softmax,AutoTokenizer,AutoModelForSequence
Classification
```

```
from transformers import AutoTokenizer,AutoModelForSequenceClassification
from scipy.special import softmax
import math
import pandas as pd
# print(data.head())
#preprocess
for i in range(500):
    twt=data[i]
    tweet_words=[]
    for word in twt.split(' '):
      if word.startswith("@") and len(word)>1:
        word=''
      elif word.startswith('http'):
        word=""
      tweet_words.append(word)
    # print(tweet_words)
    twt_pro=' '.join(tweet_words)
    #load the model and tokenizer
    rob="cardiffnlp/twitter-roberta-base-
sentiment"  #from the hugging.face
    model=AutoModelForSequenceClassification.from_pretrained(rob)
    tokenizer=AutoTokenizer.from_pretrained(rob)
    lb=['Negative','Neutral','Positive']

    #sentiment Analysis
    encoded_tweet=tokenizer(twt_pro,return_tensors='pt')
    out=model(encoded_tweet['input_ids'],encoded_tweet['attention_mask'])
    s=out[0][0].detach().numpy()
    s=softmax(s)
    mp=s[0]
    # print(s)
    # print(mp)
    j=s[0]
    k=lb[0]
      # if v%1==0:
    print("text-->",twt)
    print("label 0",lb[0],"labl 2 :",lb[1],"label :",lb[2],"\nNEG_value :
",s[0],"\nNE_value",s[1],"\nP_value :",s[2],"\n")
      # v+=1
```

Technique used for topic modelling

The process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order ***to determine whether the writer's attitude towards a particular topic, product, etc. is positive, negative, or neutral.***

The softmax function transforms each element of a collection by computing the exponential of each element divided by the sum of the exponentials of all the elements.

in many cases, the architecture you want to use can be guessed from the name or the path of the pretrained model you are supplying to the `from_pretrained()` method. AutoClasses are here to do this job for you so that you automatically retrieve the relevant model given the name/path to the pretrained weights/config/vocabulary.
Instantiating one of AutoConfig, AutoModel, and AutoTokenizer will directly create a class of the relevant architecture. For instance

```
from transformers import AutoTokenizer,AutoModelForSequenceClassification
from scipy.special import softmax
import math
import pandas as pd
# print(data.head())
#preprocess
p=0;
n=0;
ne=0;
for i in range(500):
    twt=data[i]
    tweet_words=[]
    for word in twt.split(' '):
      if word.startswith("@") and len(word)>1:
        word=''
      elif word.startswith('http'):
        word=""
      tweet_words.append(word)
    # print(tweet_words)
    twt_pro=' '.join(tweet_words)
    #load the model and tokenizer
    rob="cardiffnlp/twitter-roberta-base-sentiment"  #from the hugging.face
    model=AutoModelForSequenceClassification.from_pretrained(rob)
    tokenizer=AutoTokenizer.from_pretrained(rob)
    lb=['Negative','Neutral','Positive']

    #sentiment Analysis
    encoded_tweet=tokenizer(twt_pro,return_tensors='pt')
    out=model(encoded_tweet['input_ids'],encoded_tweet['attention_mask'])
    s=out[0][0].detach().numpy()
    s=softmax(s)
    mp=s[0]
    # print(s)
    # print(mp)
    j=s[0]
    k=lb[0]
    if s[2]>0.5:
      p=p+1
    elif s[0]>0.5:
      n=n+1
    else:
      ne=ne+1

      # if v%1==0:
    print(lb[0],lb[1],lb[2],"\n",s[0],"\n",s[1],"\n",s[2],"\n")
      # v+=1
print(p)
print(n)
print(ne)
```

The extracted data is in

TWEET.CSV

The cleaned data is in

FINALSA.CSV

Overall information and results for sentimal analysis on covid-19 in

SA2.IPYNB