# Sentiment Analysis

The method of determining whether a block of text is good, negative, or neutral is known as sentiment analysis.  Sentiment analysis is the contextual mining of words that reveals the social sentiment of a brand and aids businesses in determining if the product they are producing will find a market.

For this assignment, I attempted to train my model in two methods:

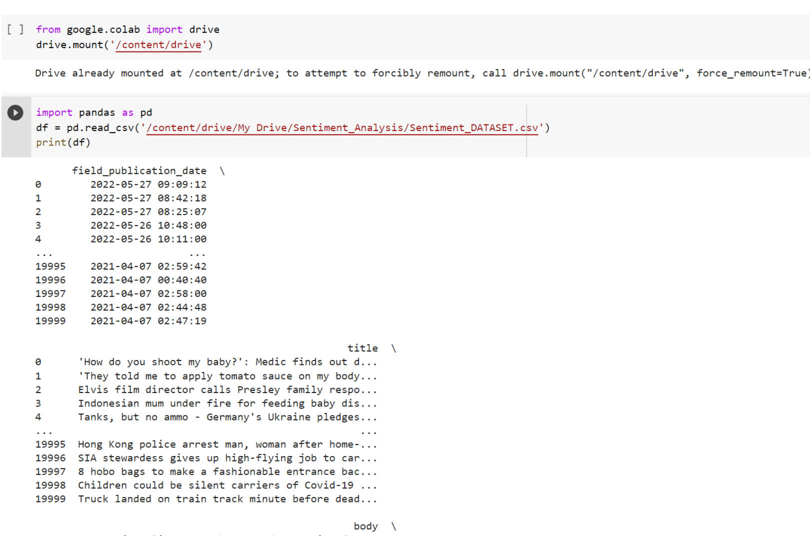
1. Rule Based Approach:
   In this attempt text mining is used to demonstrate how to use Python to compute two scores: sentiment polarity and subjectivity. It may determine whether the text contains positive or negative feedback by looking at the polarity, which ranges from -1 to 1 (negative to positive).
   [https://github.com/anumun16/Article-Sentiment-Analysis/blob/main/SentimentAnalysis.ipynb](https://github.com/anumun16/Article-Sentiment-Analysis/blob/main/SentimentAnalysis.ipynb)

2. Automatic Approach:
   This strategy utilizes the machine learning method. Predictive analysis is first performed once the datasets have been trained. Word extraction from the text is the subsequent procedure. Creating a model using MultinomialNB,GaussianNB.
   This method has been used for the **Webapp, using NLTK and deployed using Heroku**.
   [https://github.com/anumun16/Article-Sentiment-Analysis/blob/main/Sentiment%20Analysis%20Webapp.ipynb](https://github.com/anumun16/Article-Sentiment-Analysis/blob/main/Sentiment%20Analysis%20Webapp.ipynb)

- ## Rule Base Approach

Tokenization, parsing, and the lexicon technique are rule-based. The strategy counts how many positive and negative terms are present in the sample. If there are more positive words than negative words, the emotion is positive; otherwise, it is the opposite. This model was run on Google Colab.

Mounting google drive and reading the dataset.csv file:
Importing Libraries:



```
[ ] from google.colab import drive
    drive.mount('/content/drive')

    Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True)

    import pandas as pd
    df = pd.read_csv('/content/drive/My Drive/Sentiment_Analysis/Sentiment_DATASET.csv')
    print(df)

          field_publication_date  \
    0        2022-05-27 09:09:12
    1        2022-05-27 08:42:18
    2        2022-05-27 08:25:07
    3        2022-05-26 10:48:00
    4        2022-05-26 10:11:00
    ...                      ...
    19995    2021-04-07 02:59:42
    19996    2021-04-07 00:40:40
    19997    2021-04-07 02:58:00
    19998    2021-04-07 02:44:48
    19999    2021-04-07 02:47:19

                                                       title  \
    0      'How do you shoot my baby?': Medic finds out d...
    1      'They told me to apply tomato sauce on my body...
    2      Elvis film director calls Presley family respo...
    3      Indonesian mum under fire for feeding baby dis...
    4      Tanks, but no ammo - Germany's Ukraine pledges...
    ...                                                  ...
    19995  Hong Kong police arrest man, woman after home-...
    19996  SIA stewardess gives up high-flying job to car...
    19997  8 hobo bags to make a fashionable entrance bac...
    19998  Children could be silent carriers of Covid-19 ...
    19999  Truck landed on train track minute before dead...

                                                        body  \
```

```
[ ] import pandas as pd
    import re
    import string
    import numpy as np
    import random
    import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as sns
    %matplotlib inline
    from wordcloud import WordCloud
    from textblob import TextBlob
```

Data preprocessing:
The dataset underwent a number of pre-processing processes, primarily the removal of stopwords and emojis. For easier generalization, the text is then changed to lowercase.
Punctuation was then cleaned up and eliminated, which lessened the dataset's needless noise. The repetitive letters from the words were then eliminated.
For better results, I have finally done stemming (which reduces the words to their derived stems) and lemmatization (which returns the derived words to their lemma-like root form).

```
[ ] import nltk
    from nltk.stem import WordNetLemmatizer
    lemma = WordNetLemmatizer()
    nltk.download('stopwords')
    from nltk.corpus import stopwords

    [nltk_data] Downloading package stopwords to /root/nltk_data...
    [nltk_data]   Package stopwords is already up-to-date!
```

Converting text to lowercase:
The column ‘body’ and ‘title’ are made to lowercase

```
[ ] df['title']=df['title'].str.lower()
    df['title'].head()

    0    'how do you shoot my baby?': medic finds out d...
    1    'they told me to apply tomato sauce on my body...
    2    elvis film director calls presley family respo...
    3    indonesian mum under fire for feeding baby dis...
    4    tanks, but no ammo - germany's ukraine pledges...
    Name: title, dtype: object
```

```
[ ] df['body']=df['body'].str.lower()
    df['body'].head()

    0    it was just like any other tuesday morning for...
    1    singapore - when fraudsters posing as immigrat...
    2    cannes, france — film director baz luhrmann sa...
    3    parents are generally encouraged to include so...
    4    berlin — four weeks ago, germany agreed to sen...
    Name: body, dtype: object
```

## Cleaning Stopwords:

```
STOPWORDS = set(stopwords.words('english'))
def cleaning_stopwords(text):
    return " ".join([word for word in str(text).split() if word not in STOPWORDS])
df['title'] = df['title'].apply(lambda text: cleaning_stopwords(text))
df['title'].head()
## removing stopwords from column title
```

```
0    'how shoot baby?': medic finds daughter killed...
1    'they told apply tomato sauce body look injure...
2    elvis film director calls presley family respo...
3    indonesian mum fire feeding baby dish 25 chill...
4    tanks, ammo - germany's ukraine pledges show m...
Name: title, dtype: object
```

```
[ ] STOPWORDS = set(stopwords.words('english'))
def cleaning_stopwords(text):
    return " ".join([word for word in str(text).split() if word not in STOPWORDS])
df['body'] = df['body'].apply(lambda text: cleaning_stopwords(text))
df['body'].head()
## removing stopwords from column body
```

```
0    like tuesday morning angel garza, woke early d...
1    singapore - fraudsters posing immigration chec...
2    cannes, france — film director baz luhrmann sa...
3    parents generally encouraged include fibre chi...
4    berlin — four weeks ago, germany agreed send d...
Name: body, dtype: object
```

## Removing punctuation, numbers and special characters:

```
[ ] import string
english_punctuations = string.punctuation
punctuations_list = english_punctuations
def cleaning_punctuations(text):
    translator = str.maketrans('', '', punctuations_list)
    return text.translate(translator)
df['title']= df['title'].apply(lambda x: cleaning_punctuations(x))
df['title'].head()
```

```
0    how shoot baby medic finds daughter killed tex...
1    they told apply tomato sauce body look injured...
2    elvis film director calls presley family respo...
3    indonesian mum fire feeding baby dish 25 chill...
4    tanks ammo  germanys ukraine pledges show mili...
Name: title, dtype: object
```

```
import string
english_punctuations = string.punctuation
punctuations_list = english_punctuations
def cleaning_punctuations(text):
    translator = str.maketrans('', '', punctuations_list)
    return text.translate(translator)
df['body']= df['body'].apply(lambda x: cleaning_punctuations(x))
df['body'].head()
```

```
def cleaning_repeating_char(text):
    return re.sub(r'(.)1+', r'1', text)
df['body'] = df['body'].apply(lambda x: cleaning_repeating_char(x))
df['body'].head()
```

```
0    like tuesday morning angel garza woke early dr...
1    singapore  fraudsters posing immigration check...
2    cannes france — film director baz luhrmann sai...
3    parents generally encouraged include fibre chi...
4    berlin — four weeks ago germany agreed send do...
Name: body, dtype: object
```

```
[ ] def cleaning_numbers(data):
    return re.sub('[0-9]+', ' ', data)
df['body'] = df['body'].apply(lambda x: cleaning_numbers(x))
df['body'].head()
```

```
0    like tuesday morning angel garza woke early dr...
1    singapore  fraudsters posing immigration check...
2    cannes france — film director baz luhrmann sai...
3    parents generally encouraged include fibre chi...
4    berlin — four weeks ago germany agreed send do...
Name: body, dtype: object
```

```
[ ] def transform_text(text):
    return ' '.join([word for word in text.split() if len(word) > 2])
df['title'] = df['title'].apply(lambda x: transform_text(x))
df['title'].head()
```

```
0    how shoot baby medic finds daughter killed tex...
1    they told apply tomato sauce body look injured...
2    elvis film director calls presley family respo...
3    indonesian mum fire feeding baby dish chilli p...
```

Tokenization:
The strings can be divided into a list of terms using tokenization. Tokenization functions built into the Natural Language Toolkit will be used in this example. Regex can be used to tokenize it as well, although it is more challenging. Even so, you have more control over our text with this.

Stemming:
I realized sometimes, while stemming words, one might realize that trying to find roots is illogical and ludicrous. Because stemming is rule-based, it removes suffixes from words in accordance with predetermined guidelines.

```
import nltk
st = nltk.PorterStemmer()
def stemming_on_text(data):
    text = [st.stem(word) for word in data]
    return data
df['body']= df['body'].apply(lambda x: stemming_on_text(x))
df['body'].head()
```

```
0    [like, tuesday, morning, angel, garza, woke, e...
1    [singapore, fraudsters, posing, immigration, c...
2    [cannes, france, film, director, baz, luhrmann...
3    [parents, generally, encouraged, include, fibr...
4    [berlin, four, weeks, ago, germany, agreed, se...
Name: body, dtype: object
```

Lemmanization:
Finding the linked word's lexical form through lemmatization is a method. Stemming is distinct from it. Compared to stemming, the calculation method is more time-consuming.
Similar to stemming, lemmatization aims to reduce inflectional forms to a basic form. It does not just remove inflections like stemming does. Instead, it makes use of lexical knowledge bases to obtain the proper word base forms.

```
[ ] lm = nltk.WordNetLemmatizer()
    def lemmatizer_on_text(data):
        text = [lm.lemmatize(word) for word in data]
        return data
    df['body'] = df['body'].apply(lambda x: lemmatizer_on_text(x))
    df['body'].head()
```

```
0    [like, tuesday, morning, angel, garza, woke, e...
1    [singapore, fraudsters, posing, immigration, c...
2    [cannes, france, film, director, baz, luhrmann...
3    [parents, generally, encouraged, include, fibr...
4    [berlin, four, weeks, ago, germany, agreed, se...
Name: body, dtype: object
```

Calculating Subjectivity and Polarity

```python
def getSubjectivity(text):
    return TextBlob('text').sentiment.subjectivity

#create a function to get the polarity
def getpolarity(text):
    return TextBlob('text').sentiment.polarity

#create two new columns
df['subjectivity']=df['body'].apply(getSubjectivity)
df['polarity']=df['body'].apply(getpolarity)

#show the new datafraem with the new columns
df.head()
```

| | field_publication_date | title | body | path | catagory | subjectivity | polarity | analysis |
|---|---|---|---|---|---|---|---|---|
| 0 | 2022-05-27 09:09:12 | [how, shoot, baby, medic, finds, daughter, kil... | [like, tuesday, morning, angel, garza, woke, e... | /world/how-do-you-shoot-my-baby-medic-finds-ou... | world | 0.0 | 0.0 | neutral |
| 1 | 2022-05-27 08:42:18 | [they, told, apply, tomato, sauce, body, look,... | [singapore, fraudsters, posing, immigration, c... | /singapore/they-told-me-apply-tomato-sauce-my-... | singapore | 0.0 | 0.0 | neutral |
| 2 | 2022-05-27 08:25:07 | [elvis, film, director, calls, presley, family... | [cannes, france, film, director, baz, luhrmann... | /entertainment/elvis-film-director-calls-presl... | entertainment | 0.0 | 0.0 | neutral |
| 3 | 2022-05-26 10:48:00 | [indonesian, mum, fire, feeding, baby, dish, c... | [parents, generally, encouraged, include, fibr... | /asia/indonesian-mum-under-fire-feeding-baby-d... | asia | 0.0 | 0.0 | neutral |
| 4 | 2022-05-26 10:11:00 | [tanks, ammo, germanys, ukraine, pledges, show... | [berlin, four, weeks, ago, germany, agreed, se... | /world/tanks-no-ammo-germanys-ukraine-pledges-... | world | 0.0 | 0.0 | neutral |

It was observed that on analysing the data all values were neutral. Which is a matter of concern since not all the data should be neutral.
Hence to compare it with Automatic approach a conclusion is provided at the end of this documentation.

```python
def getAnalysis(score):
    if score<0:
        return 'negative'
    elif score==0:
        return 'neutral'
    else:
        return 'positive'

df['analysis']=df['polarity'].apply(getAnalysis)

#show dataFrame
df.head()
```

| | field_publication_date | title | body | path | catagory | subjectivity | polarity | analysis |
|---|---|---|---|---|---|---|---|---|
| 0 | 2022-05-27 09:09:12 | [how, shoot, baby, medic, finds, daughter, kil... | [like, tuesday, morning, angel, garza, woke, e... | /world/how-do-you-shoot-my-baby-medic-finds-ou... | world | 0.0 | 0.0 | neutral |
| 1 | 2022-05-27 08:42:18 | [they, told, apply, tomato, sauce, body, look,... | [singapore, fraudsters, posing, immigration, c... | /singapore/they-told-me-apply-tomato-sauce-my-... | singapore | 0.0 | 0.0 | neutral |
| 2 | 2022-05-27 08:25:07 | [elvis, film, director, calls, presley, family... | [cannes, france, film, director, baz, luhrmann... | /entertainment/elvis-film-director-calls-presl... | entertainment | 0.0 | 0.0 | neutral |
| 3 | 2022-05-26 10:48:00 | [indonesian, mum, fire, feeding, baby, dish, c... | [parents, generally, encouraged, include, fibr... | /asia/indonesian-mum-under-fire-feeding-baby-d... | asia | 0.0 | 0.0 | neutral |
| 4 | 2022-05-26 10:11:00 | [tanks, ammo, germanys, ukraine, pledges, show... | [berlin, four, weeks, ago, germany, agreed, se... | /world/tanks-no-ammo-germanys-ukraine-pledges-... | world | 0.0 | 0.0 | neutral |

- ## Automatic Approach

This strategy utilizes the machine learning method. Predictive analysis is first performed once the datasets have been trained. Word extraction from the text is the subsequent procedure. Different methods, including Naive Bayes, Linear Regression, Support Vector, and Deep Learning, can be used to extract text, just like these machine learning techniques.

This model was run on Jupyter Notebook and the webapp was deployed on Heroku.

**First the model is created using NLTK and MAchine Learning.**

Importing Libraries:

In [1]:
```
import numpy as np ## scientific computation
import pandas as pd ## loading dataset file
import matplotlib.pyplot as plt ## Visulization
import nltk  ## Preprocessing Reviews
nltk.download('stopwords') ##Downloading stopwords
from nltk.corpus import stopwords ## removing all the stop words
from nltk.stem.porter import PorterStemmer ## stemming of words
import re  ## To use Regular expression
```

```
[nltk_data] Downloading package stopwords to C:\Users\Anil
[nltk_data]     munde\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Adding path to the jupyter model:

In [2]:
```
df = pd.read_csv('C:\\Users\\Anil munde\\OneDrive\\Desktop\\Neural Lab\\Sentiment Analysis Webapp\\Sentiment_DATASET.csv')
df.head()
```

Out[2]:

| | field_publication_date | title | body | path | catagory |
|---|---|---|---|---|---|
| 0 | 2022-05-27 09:09:12 | 'How do you shoot my baby?': Medic finds out d... | It was just like any other Tuesday morning for... | /world/how-do-you-shoot-my-baby-medic-finds-ou... | world |
| 1 | 2022-05-27 08:42:18 | 'They told me to apply tomato sauce on my body... | SINGAPORE - When fraudsters posing as Immigrat... | /singapore/they-told-me-apply-tomato-sauce-my-... | singapore |
| 2 | 2022-05-27 08:25:07 | Elvis film director calls Presley family respo... | CANNES, France — Film director Baz Luhrmann sa... | /entertainment/elvis-film-director-calls-presl... | entertainment |
| 3 | 2022-05-26 10:48:00 | Indonesian mum under fire for feeding baby dis... | Parents are generally encouraged to include so... | /asia/indonesian-mum-under-fire-feeding-baby-d... | asia |
| 4 | 2022-05-26 10:11:00 | Tanks, but no ammo - Germany's Ukraine pledges... | BERLIN — Four weeks ago, Germany agreed to sen... | /world/tanks-no-ammo-germanys-ukraine-pledges-... | world |

Cleaning the Dataset:
To save the cleaned version of data an empty list is created. Anything in a sentence can be changed to anything by a loop from 0 to 20000. All of the punctuation has been replaced with blank white space, and data conversion to lowercase.
Creating a list of terms by dividing the input making an item for the porterstemmer class ().
Taking not out of the stop words will make it simpler to distinguish between positive and negative terms. Running a loop to determine the sentence's length, checking each word's stopword status, applying stemming to the text, and adding the results to the list.

```
In [5]: corpus = []
for i in range(0,20000):   #20000 articles
     senta = re.sub('[^a-zA-Z]'," ",df["body"][i])
     senta = senta.lower()
     senta = senta.split()
     pe = PorterStemmer()
     all_stopword = stopwords.words('english')
     all_stopword.remove('not')
     senta = [pe.stem(word) for word in senta if not word in set(all_stopword)]
     senta = " ".join(senta)
     corpus.append(senta)
print(corpus)
```

```
['like tuesday morn angel garza woke earli drop daughter ameri jo garza school would turn last time father would ever get see
hold year old morn may year old gunman open fire classroom elementari student robb elementari school uvald texa leav children
two adult dead garza medic aid call scene assist mass shoot aftermath see littl girl cover blood head toe went ask okay garza
relat interview cnn anderson cooper live air wednesday http www tiktok com nowthi video hyster say shot best friend kill best
friend not breath tri call cop recount garza shoulder start shake uncontrol interview ask littl girl name said told said amer
i garza heart drop turn best friend none daughter interview father broke take second compos ad ameri sweetest littl girl noth
wrong ameri jo garza kill gunman open fire robb elementari school photo screengrab facebook laura garza want know victim ad g
arza fought back tear look girl shoot oh babi shoot babi garza later continu interview sob hug photo daughter win honour awar
d ameri grandmoth berlinda iren arreola told news site daili beast granddaught fatal shot midst attempt dial phone garza mour
n daughter facebook also show appreci tri find daughter whereabout littl love fli high angel garza wrote facebook also took t
ime remind other import messag pleas take second grant hug famili tell love ameri mother kimberli garcia post facebook never
comprehend loss daughter not deserv sweet babi girl wrote suppos live life without never understand love never ever ameri one
peopl kill one deadliest school shoot us histori sinc children adult kill sandi hook elementari school newton connecticut reu
ter report news agenc also ad gunman post intent social media beforehand barg unchalleng unlock door kill children includ ame
ri two teacher hole classroom hour tactic team storm kill school district uvald stand polici lock entranc includ classroom do
or safeti precaut one student told reuter door left unlock day shoot allow visit parent come go award day event investig stil
l seek motiv massacr also read children carri id texa offici face horror identifi bodi knew aishahm asiaon com', 'singapor fr
audster pose immigr checkpoint author ica offic accus year old peter not real name april involv scam teenag terrifi said mobi
l number china use cheat victim singapor prove innoc teenag china nation told deposit yuan one scammer bank account peter fai
l rais sum agre fear cooper scammer scheme fake kidnap peter guardian singapor jenni not real name alert polic may not return
```

By setting the maximum features of an object for the count vectorizer to 20000, only retrieving the first 20000 columns. We are fitting our corpus, turning it into vectors, then integrating it with X using CV, integrating y with column values.

```
: from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features=20000) ##20000 columns
X = cv.fit_transform(corpus).toarray()
y = df["body"]
```

```
: cls.score(X_test,y_test)
```

```
: classifier.score(X_test,y_test)
```

```
: y_pred = cls.predict(X_test)
type(y_test)
```

```
: print(np.concatenate((y_pred.reshape(len(y_pred),1), np.array(y_test).reshape(len(y_test),1)),1))
```

```
: from sklearn.metrics import confusion_matrix,accuracy_score
cm = confusion_matrix(y_test, y_pred)
score = accuracy_score(y_test,y_pred)
print(cm,score*100)
```

Using Naive Bayes Theorem, we can observe the accuracy of the model as 77.5% is higher.

```
Article Sentiment Analysis
  |____templates
        |___index.html  ## homepage file
        |___result.html ## to show prediction
  |____static
        |____style.css  ## css file
  |____app.py  ## main flask file
```

## Conclusion

- It is very challenging to determine whether a sentence is optimistic or pessimistic when the data is presented in the form of a tone.
- You must determine if the data is beneficial or negative if it is shown as an emoji.
- A neutral statement is difficult to compare.
- Automatic Approach proved to be better than Rule based Approach.
- Words with strong positive (+1) and negative (-1) polarity scores include "love" and "hate." These are simple to comprehend. However, there are word conjugations that fall in the middle of the polarity spectrum, such as "not so awful," which can also indicate "average" (-75). These kinds of sentences are occasionally omitted, which lowers the sentiment score.
- "jupyter notebook --NotebookApp.iopub_data_rate_limit=1.0e10" statement was used to resolve the IOPub data rate exceeded.

Resources:

https://stackoverflow.com/questions/65172293/how-to-solve-iopub-data-rate-exceeded-in-jupyter-notebook

https://www.datacamp.com/tutorial/text-analytics-beginners-nltk

https://www.geeksforgeeks.org/what-is-sentiment-analysis/

https://github.com/priyansh19/Suggestion_Mining_Using_Twitter_Data

https://towardsdatascience.com/stemming-lemmatization-what-ba782b7c0bd8