

```
In [1]: import re, string
import pandas as pd
# plotting
import seaborn as sns
import matplotlib.pyplot as plt
# nltk
from nltk import word_tokenize, sent_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer, WordNetLemmatizer
# sentiment
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
# sklearn
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA, TruncatedSVD
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
# from sklearn.svm import LinearSVC
# from sklearn.naive_bayes import BernoulliNB
# from sklearn.linear_model import LogisticRegression
# from sklearn.model_selection import train_test_split
# from sklearn.metrics import confusion_matrix, classification_report
```

## DataFrame Analysis

DataSet Link: [Kaggle](#)

```
In [2]: # read file
df = pd.read_csv('tweets/data_science.csv', engine='python')

# extract id, created_at, username, tweet
df = df[["id", "created_at", "username", "tweet"]]
df.columns = ["id", "date", "user", "tweet"]

# and convert date
df.date = pd.to_datetime(df.date, format="%Y-%m-%d %H:%M:%S IST").dt.tz_localize('EST').dt.tz_convert('Asia/Kolkata')
```

```
In [3]: # show structure
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 241386 entries, 0 to 241385
Data columns (total 4 columns):
#   Column   Non-Null Count  Dtype
---  -
0    id       241386 non-null  int64
1    date     241386 non-null  datetime64[ns, Asia/Kolkata]
2    user     241386 non-null  object
3    tweet    241386 non-null  object
dtypes: datetime64[ns, Asia/Kolkata](1), int64(1), object(2)
memory usage: 7.4+ MB
```

```
In [4]: df.sample(5)
```

Out[4]:

	id	date	user	tweet
24879	1326572031077462017	2020-11-12 09:06:33+05:30	kirkdborne	Proof of concept is old news! Showing #IoT Pro...
197459	646353498573901825	2015-09-23 08:00:57+05:30	pinnacle	Unwrap our betting focused R packages and impr...
228709	388628319639728128	2013-10-12 03:33:06+05:30	data_nerd	What can your data do for you? <a href="http://t.co/ty...">http://t.co/ty...</a>
116805	1011047378508505089	2018-06-25 16:44:14+05:30	kirkdborne	Digital Experience and Artificial Intelligence...
204413	606438980276633600	2015-06-05 04:34:54+05:30	cheltfestivals	What if...data science can solve our future? F...

## Analyze Text

```
In [5]: # preview
df.tweet = df.tweet.str.lower()

df.tweet
```

```
Out[5]: 0      what can be done? - never blindly trust an ab...
1      "we need a paradigm shift from model-centric t...
2      using high-resolution satellite data and compu...
3      .@stephenson_data shares four steps that will ...
4      "curricula is inherently brittle in a world wh...
...
241381  cda jobs data, dec: employment rose in health,...
241382  rt @filiber: have a computer science backgroun...
241383  @pop17 heck with science. i've got empirical d...
241384  all in the....data rt @noahwg dr. petra provid...
241385  "the world of retail will always be a mix of a...
Name: tweet, Length: 241386, dtype: object
```

### Cleaning and removing the stop words from the tweet text

```
In [6]: stop_words = stopwords.words('english')

def cleaning_stopwords(text: str) → str:
    return " ".join(word for word in text.split() if word not in stop_words)

df.tweet = df.tweet.apply(cleaning_stopwords)

df.tweet.head(5)
```

```
Out[6]: 0      done? - never blindly trust abstract, press re...
1      "we need paradigm shift model-centric data-cen...
2      using high-resolution satellite data computer ...
3      .@stephenson_data shares four steps help new d...
4      "curricula inherently brittle world in-demand ...
Name: tweet, dtype: object
```

### Cleaning and removing punctuations

```
In [7]: punctuations_list = string.punctuation

def cleaning_punctuations(text: str) → str:
    translator = str.maketrans('', '', punctuations_list)
    return text.translate(translator)

df.tweet = df.tweet.apply(cleaning_punctuations)

df.tweet.head(5)
```

```
Out[7]: 0    done  never blindly trust abstract press relea...
        1    we need paradigm shift modelcentric datacentri...
        2    using highresolution satellite data computer a...
        3    stephensondata shares four steps help new data...
        4    curricula inherently brittle world indemand sk...
        Name: tweet, dtype: object
```

## Cleaning and removing URL's

```
In [8]: pattern, replacement = '((www.[^s]+)|(https?://[^\s]+))', ' '

def cleaning_URLs(text: str) → str:
    return re.sub(pattern, replacement, text)

df.tweet = df.tweet.apply(cleaning_URLs)

df.tweet.head(5)
```

```
Out[8]: 0    done  never blindly trust abstract press relea...
        1    we need paradigm shift modelcentric datacentri...
        2    using highresolution satellite data computer a...
        3    stephensondata shares four steps help new data...
        4    curricula inherently brittle world indemand sk...
        Name: tweet, dtype: object
```

## Cleaning and removing Numeric numbers

```
In [9]: pattern, replacement = '[0-9]+', ''

def cleaning_numbers(text: str) → str:
    return re.sub(pattern, replacement, text)

df.tweet = df.tweet.apply(cleaning_numbers)

df.tweet.head(5)
```

```
Out[9]: 0    done  never blindly trust abstract press relea...
        1    we need paradigm shift modelcentric datacentri...
        2    using highresolution satellite data computer a...
        3    stephensondata shares four steps help new data...
        4    curricula inherently brittle world indemand sk...
        Name: tweet, dtype: object
```

## Getting tokenization of tweet text

```
In [10]: df.tweet = df.tweet.apply(word_tokenize)

df.tweet.head(5)
```

```
Out[10]: 0    [done, never, blindly, trust, abstract, press,...
        1    [we, need, paradigm, shift, modelcentric, data...
        2    [using, highresolution, satellite, data, compu...
        3    [stephensondata, shares, four, steps, help, ne...
        4    [curricula, inherently, brittle, world, indema...
        Name: tweet, dtype: object
```

## Applying Lemmatizer

```
In [11]: lemmatizer = WordNetLemmatizer()

def lemmatizer_on_text(text: str) -> str:
    return " ".join(lemmatizer.lemmatize(word) for word in text)

df.tweet = df.tweet.apply(lemmatizer_on_text)

df.tweet.head(5)
```

```
Out[11]: 0    done never blindly trust abstract press releas...
1    we need paradigm shift modelcentric datacentri...
2    using highresolution satellite data computer a...
3    stephensondata share four step help new data s...
4    curriculum inherently brittle world indemand s...
Name: tweet, dtype: object
```

## Text Sentiment Analysis

```
In [12]: analyser = SentimentIntensityAnalyzer()

def calculate_sentiment_analyser(text: str) -> dict:
    return analyser.polarity_scores(text)

def calculate_compound_score(sentiment: dict) -> float:
    return sentiment['compound']

def calculate_compound_score_sentiment(compound_score: float) -> str:
    return 'Negative' if (compound_score <= -0.05) else \
           'Positive' if (compound_score >= 0.05) else \
           'Neutral'
```

### Calculating sentiments

```
In [13]: df["sentiment_analyser"] = df.tweet.apply(calculate_sentiment_analyser)

df['compound_score'] = df.sentiment_analyser.apply(calculate_compound_score)

df['compound_score_sentiment'] = df.compound_score.apply(calculate_compound_score_sentiment)

df.head(5)
```

```
Out[13]:
```

	id	date	user	tweet	sentiment_analyser	compound_score
0	1406400408545804288	2021-06-20 15:56:01+05:30	ballouxfrancois	done never blindly trust abstract press releas...	{'neg': 0.231, 'neu': 0.629, 'pos': 0.141, 'co...	-0.4592
1	1406390341176016897	2021-06-20 15:16:01+05:30	tdatascience	we need paradigm shift modelcentric datacentri...	{'neg': 0.135, 'neu': 0.692, 'pos': 0.173, 'co...	0.0000
2	1406386311481774083	2021-06-20 15:00:00+05:30	sciencenews	using highresolution satellite data computer a...	{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound...	0.0000
3	1406383545153638402	2021-06-20 14:49:01+05:30	tdatascience	stephensondata share four step help new data s...	{'neg': 0.0, 'neu': 0.552, 'pos': 0.448, 'comp...	0.7430
4	1406358632648818689	2021-06-20 13:10:01+05:30	tdatascience	curriculum inherently brittle world indemand s...	{'neg': 0.0, 'neu': 0.895, 'pos': 0.105, 'comp...	0.4019

```
In [14]: df.compound_score_sentiment.value_counts()
```

```
Out[14]: Positive      123418  
Neutral      94702  
Negative      23266  
Name: compound_score_sentiment, dtype: int64
```

## Implementing KMeans

```
In [15]: # Considering 3 grams and minimum frq as 0  
# tf_idf_vect = TfidfVectorizer(analyzer = 'word', ngram_range = (1, 3), min_df = 0, stop_words = 'e  
tf_idf_vect = CountVectorizer(analyzer='word', ngram_range=(1,1), stop_words='english', min_df = 0.000  
tf_idf_vect.fit(df.tweet)  
desc_matrix = tf_idf_vect.transform(df.tweet)
```

```
In [16]: # implement kmeans  
num_clusters = 3  
km = KMeans(n_clusters=num_clusters)  
km.fit(desc_matrix)  
clusters = km.labels_.tolist()
```

```
In [17]: # create DataFrame films from all of the input files.  
tweets = {'Tweet': df.tweet.tolist(), 'Cluster': clusters}  
frame = pd.DataFrame(tweets, index = [clusters])  
  
frame.Cluster.value_counts()
```

```
Out[17]: 2      116392  
1       76141  
0       48853  
Name: Cluster, dtype: int64
```

```
In [18]: #create pie chart  
colors = sns.color_palette('pastel')[0:3]  
_ = plt.pie(frame.Cluster.value_counts(), labels = ["Positive", "Neutral", "Negative"], colors = col
```

