Contents

List of Figures

Text Sentimental Analysis using ML

Name: Engr. Ahmad Ali

Company: Rex Technologies

I. Problem Statement: -

The Objective of this task is to perform Sentimental Analysis on the Data which we have to get from the online platform KAGGLE, the platform where we get any type of data for our Model Training.

II. Introduction: -

In this modern era the world is moving towards the automation of things for their ease. Everything we see today from our Home to Shopping, is automated through Technology. It has vital role in our daily life, the social media platforms and use of technology is increase day after the day. It's not wrong to say we are now depending on technology for our daily work. The peak of technology is after the 20's century in which people introduce new methods and techniques to solve the human problems. Artificial Intelligence is one of them which solve many Human problems which seems to be difficult. Now days AI play its role in almost all the sectors like Medical, Retail, Electrical and Social Media. It analyzes the human behavior through Deep Learning and perform task which human could not do. Through the availability of data, the Computer is trained by humans and then that train model performs certain tasks which if human do will take many time. The analysis of the emotion or to predict a person feeling is difficult for humans some time because we can't judge perfectly about the others meanings perfectly sometimes so here comes the AI which solve this problem for humans.

III. Sentimental Analysis(SA): -

The first question arises is what is actually a sentimental analysis? It is the recognition of the Positive, Negative and Neutral Pattern or Words in the chat or Text. Many Social Media platforms or E-Commerce Website use this analysis to predict the thought of Consumer's. It is difficult for Humans to **read all chat's** and then predict the status about that Chat. Nobody have that plenty of time. So in order to save **time** and in order to **improve** their Business Companies Invest in Technologies like AI or IOT. In order to perform Sentimental Analysis, we have to use AI Build-In Modules or Methods like NLP, Transformers, ML Models like Logistic Regression Support-Vector-Machines and Deep Learning. In Text analysis or Sentimental Analysis our main goal is to extract those words which show the feelings of a person in text such as if a person text in tweet that **"I am Happy today because I pass the Exam"** So here **Happy** is our main target which show that Text has **Positive Meaning**.

IV. Steps in Sentimental Analysis: -

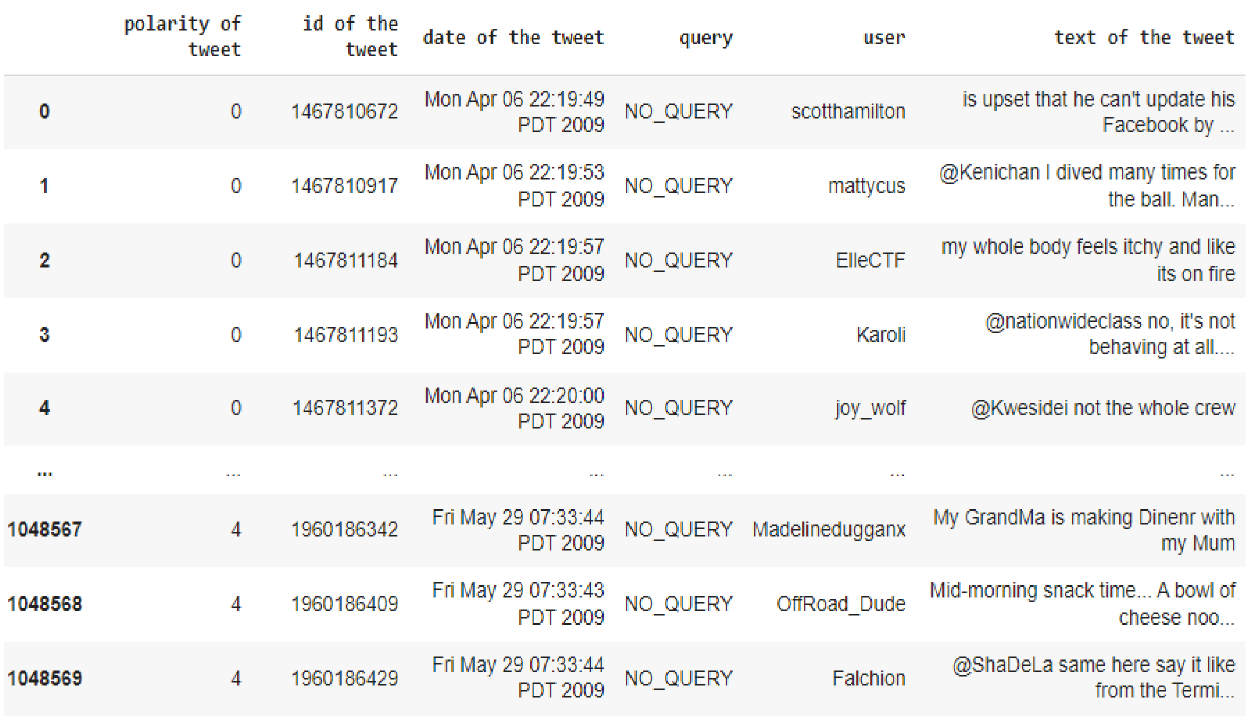
- Getting Data and Cleaning

- Visualize Data(EDA)
- Tokenization
- Stemming
- Stop Wards
- Feature Extraction through TF-IDF
- Model Selection
- Training Model
- Testing Model

V. Getting Data and Cleaning Data: -

That step is the most Crucial step for any kind of problem or Model Training. All the other steps are depending on this step. We get our problem set data from the KAGGLE and then clean the data. The selection of data involves following steps:

- Identify Problem
- Choose Data according to that problem
- Clean the data (Remove duplicated, Fill Null values with Mode, Mean etc.)
- Removing Punctuation and number's (Text Analysis case)
- Visualize Data
- Correlation hunting



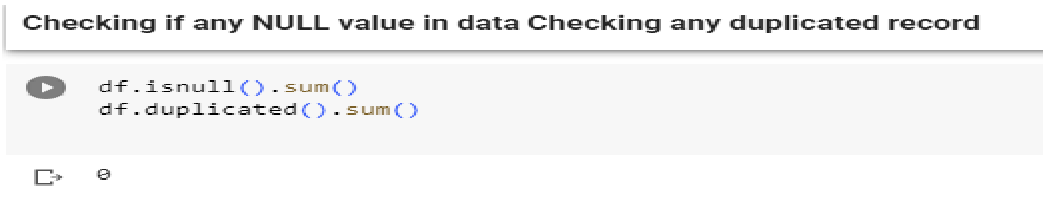
	polarity of tweet	id of the tweet	date of the tweet	query	user	text of the tweet
0	0	1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	scotthamilton	is upset that he can't update his Facebook by ...
1	0	1467810917	Mon Apr 06 22:19:53 PDT 2009	NO_QUERY	mattycus	@Kenichan I dived many times for the ball. Man...
2	0	1467811184	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	ElleCTF	my whole body feels itchy and like its on fire
3	0	1467811193	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	Karoli	@nationwideclass no, it's not behaving at all....
4	0	1467811372	Mon Apr 06 22:20:00 PDT 2009	NO_QUERY	joy_wolf	@Kwesidei not the whole crew
...	...	...	...	...	...	...
1048567	4	1960186342	Fri May 29 07:33:44 PDT 2009	NO_QUERY	Madelinedugganx	My GrandMa is making Dinenr with my Mum
1048568	4	1960186409	Fri May 29 07:33:43 PDT 2009	NO_QUERY	OffRoad_Dude	Mid-morning snack time... A bowl of cheese noo...
1048569	4	1960186429	Fri May 29 07:33:44 PDT 2009	NO_QUERY	Falchion	@ShaDeLa same here say it like from the Termi...

Figure 1 Data Use for Sentimental Analysis

The data which we need for our problem is Text Base data and that Data is the tweet data for many users which is basically a Social Media base data which is shown in figure 1 above. It has 6 Columns in which there is a,

- ID of User
- Polarity of tweet(Sentiment)
- User Name
- Query
- Date of Tweet
- Text of Tweet

During the Cleaning I first check whether there is any NULL value or Duplicates in the data through the following command,



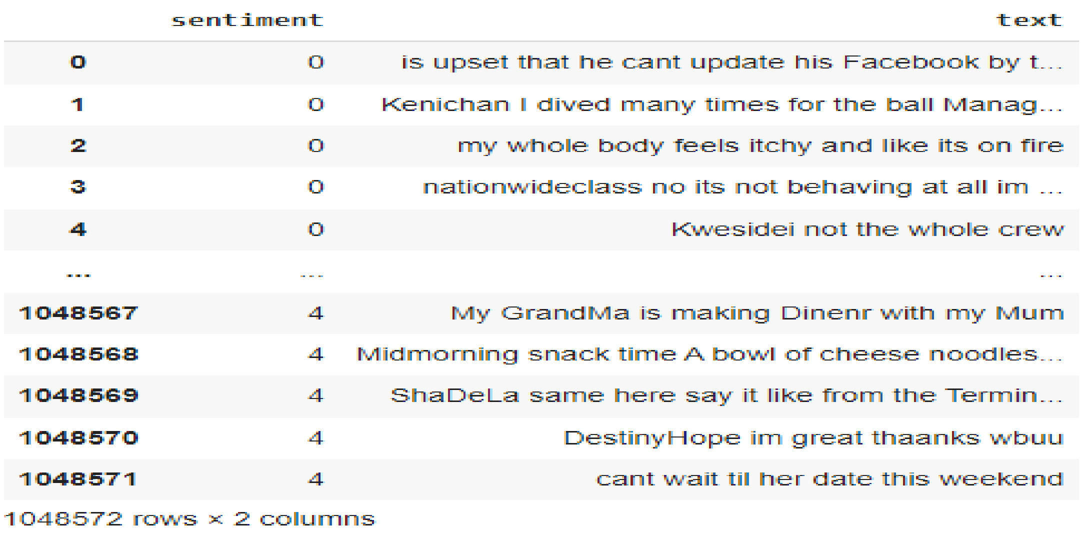
Checking if any NULL value in data Checking any duplicated record

```
df.isnull().sum()
df.duplicated().sum()
```

```
0
```

Figure 2 Checking Null and Duplication

I found no NULL values and Duplicates in my data so next I remove all the irrelevant Columns from my data and remove all the numbers and punctuations from data in order to clean it.



	sentiment	text
0	0	is upset that he cant update his Facebook by t...
1	0	Kenichan I dived many times for the ball Manag...
2	0	my whole body feels itchy and like its on fire
3	0	nationwideclass no its not behaving at all im ...
4	0	Kwesidei not the whole crew
...	...	...
1048567	4	My GrandMa is making Dinenr with my Mum
1048568	4	Midmorning snack time A bowl of cheese noodles...
1048569	4	ShaDeLa same here say it like from the Termin...
1048570	4	DestinyHope im great thaanks wbuu
1048571	4	cant wait til her date this weekend

1048572 rows × 2 columns

Figure 3 Data After Cleaning

VI. Visualization(EDA): -

In this process we have to Visualize our data which give us an insight knowledge about our data. It is not possible for humans to read all the rows and columns of data set. So to make it easy we use Graphs to see the behavior or pattern in the data. It helps us in following,

- Data Exploration
- Feature Selection for model
- Outlier Detection
- Model Selection
- Hyper-parameter tuning for model
- Pattern Recognition

We use **Words Cloud** method and count plot in order to visualize my data. Word Cloud is a tool or method to visualize the text data. It gives the image of words which occur most frequently in our text. We use that method for both Positive and Negative sentiments and see what words are appearing most. Second graph is count plot which give plot of total count for our 2 classes.

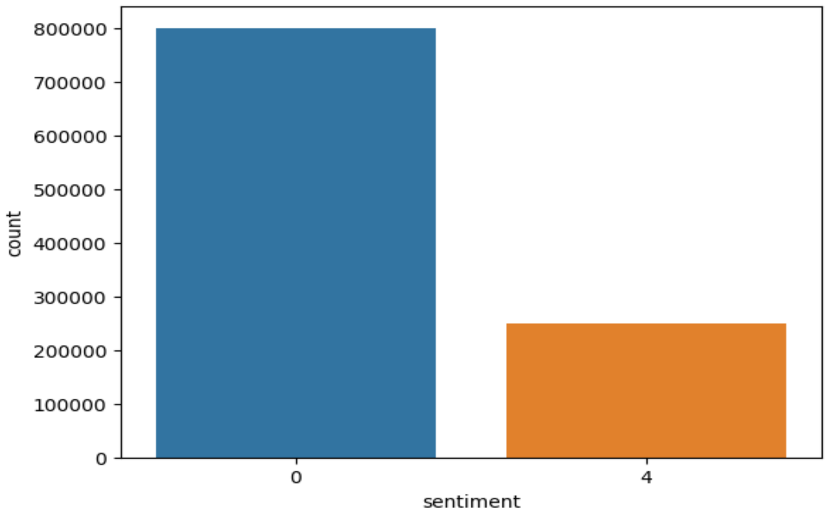


Figure 4 Count plot for Sentiments

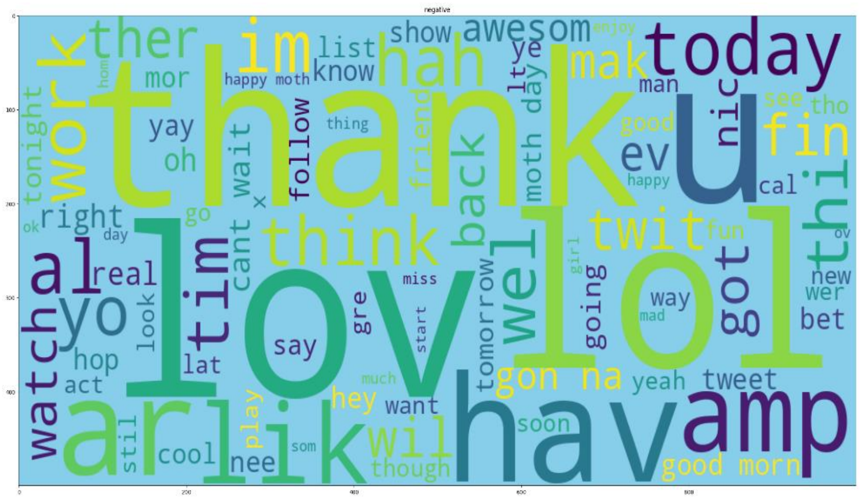


Figure 5 Image for Positive Words

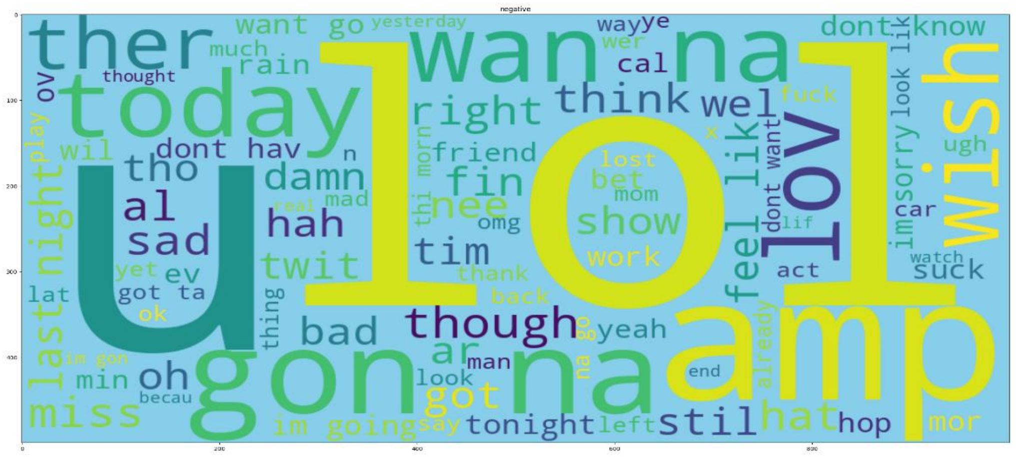
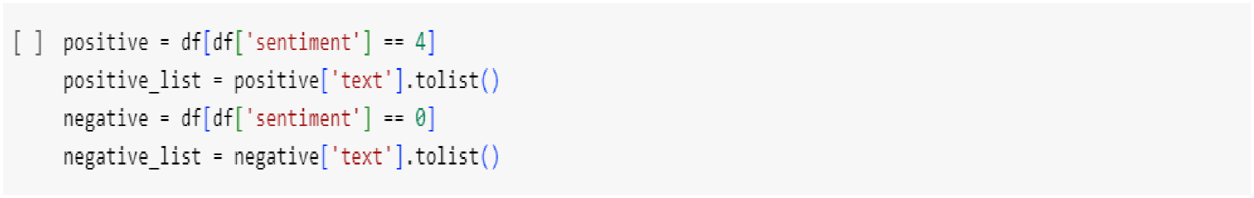
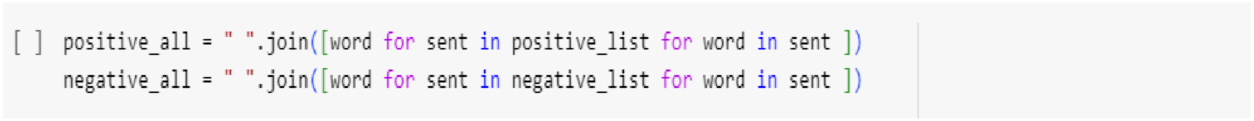


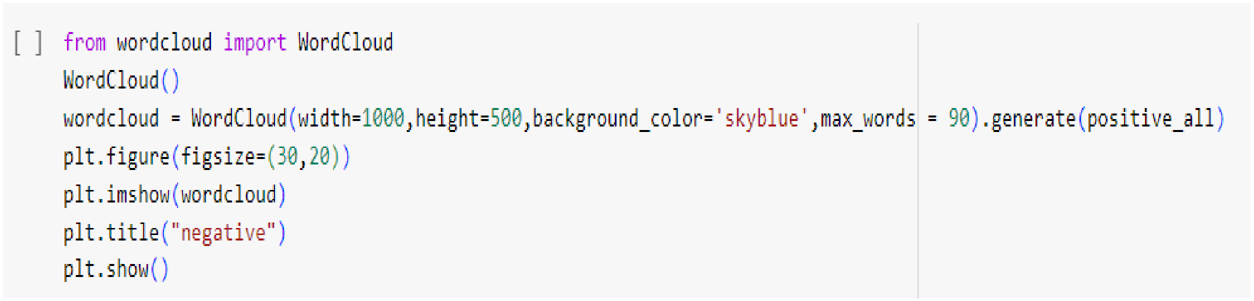
Figure 6 Image for Negative Words



```
[ ]  positive = df[df['sentiment'] == 4]
     positive_list = positive['text'].tolist()
     negative = df[df['sentiment'] == 0]
     negative_list = negative['text'].tolist()
```



```
[ ]  positive_all = " ".join([word for sent in positive_list for word in sent ])
     negative_all = " ".join([word for sent in negative_list for word in sent ])
```



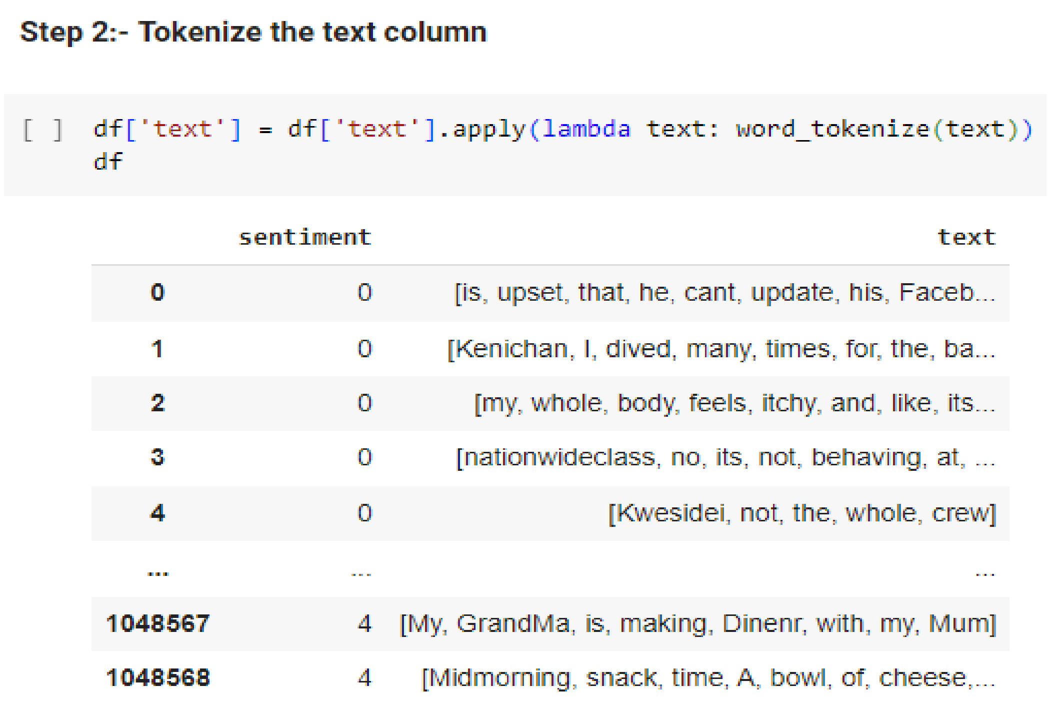
```
[ ]  from wordcloud import WordCloud
     WordCloud()
     wordcloud = WordCloud(width=1000,height=500,background_color='skyblue',max_words = 90).generate(positive_all)
     plt.figure(figsize=(30,20))
     plt.imshow(wordcloud)
     plt.title("negative")
     plt.show()
```

```
from wordcloud import WordCloud
WordCloud()
wordcloud = WordCloud(width=1000,height=500,background_color='skyblue'
  ,max_words = 90).generate(negative_all)
plt.figure(figsize=(30,20))
plt.imshow(wordcloud)
plt.title("negative")
plt.show()
```

Figure 7 Code for EDA

VII. Tokenization on Text: -

A key text processing task in natural language processing (NLP) is tokenization. It entails dividing a text into tokens, which are shorter pieces of text. Depending on the level of granularity necessary for a certain NLP activity, tokens can be individual words, sub words, or even characters. Tokenization is a preprocessing procedure used in a variety of NLP applications to facilitate working with text input and it also help the Computer to understand the Word or Sentence. Machine assign a specific number to each word which help in processing. It also helps in counting frequency of the words in the text. I apply Tokenization to my data as shown below,



Step 2:- Tokenize the text column

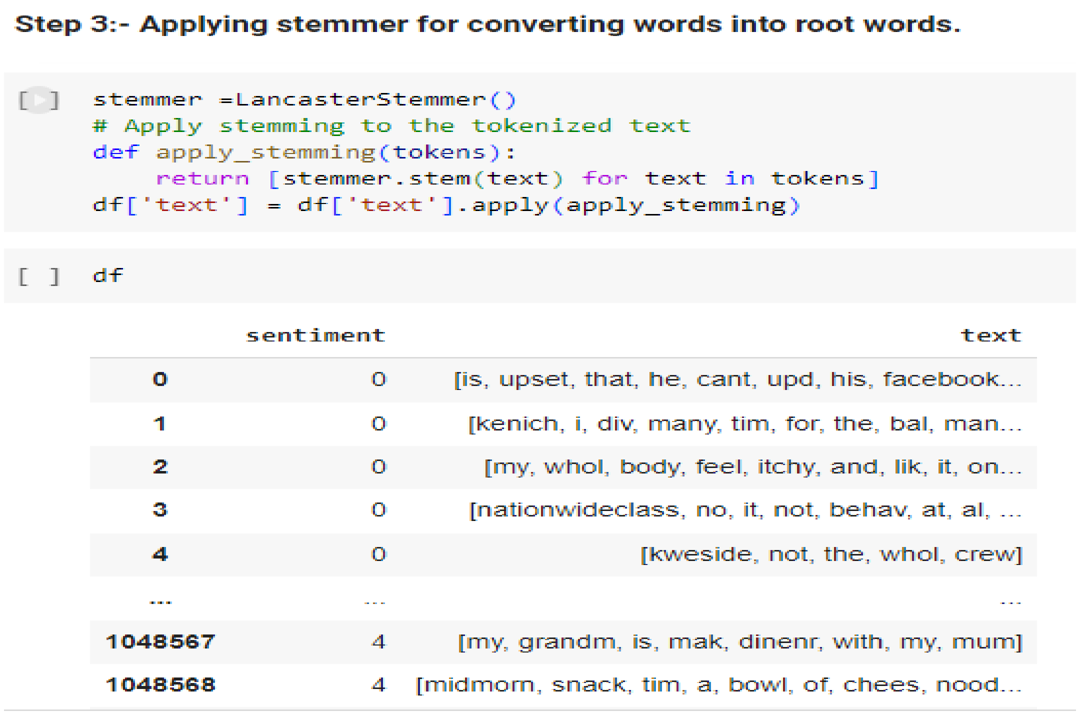
```
[ ] df['text'] = df['text'].apply(lambda text: word_tokenize(text))
    df
```

	sentiment	text
0	0	[is, upset, that, he, cant, update, his, Faceb...
1	0	[Kenichan, I, dived, many, times, for, the, ba...
2	0	[my, whole, body, feels, itchy, and, like, its...
3	0	[nationwideclass, no, its, not, behaving, at, ...
4	0	[Kwesidei, not, the, whole, crew]
...	...	...
1048567	4	[My, GrandMa, is, making, Dinenr, with, my, Mum]
1048568	4	[Midmorning, snack, time, A, bowl, of, cheese,...

Figure 8 Data after Tokenization

VIII. Stemming: -

Stemming also called Lemmatization is the Process to Normalize the words or text in to its **root word**. For example, if we have these words in our long text Affected, Affects, Affection and Affectation then these all word has same **Root** word which is **Affect.** So Stemming Perform that Task on out text and it also reduce the Length of our text which has no effect on model. The Main goal is to just get the Most important words which show sentiments or felling of a person. Stemming perform common prefixes and suffixes on the text or words to find the root words.


Step 3:- Applying stemmer for converting words into root words.

```
[ ]  stemmer =LancasterStemmer()
     # Apply stemming to the tokenized text
     def apply_stemming(tokens):
         return [stemmer.stem(text) for text in tokens]
     df['text'] = df['text'].apply(apply_stemming)
```

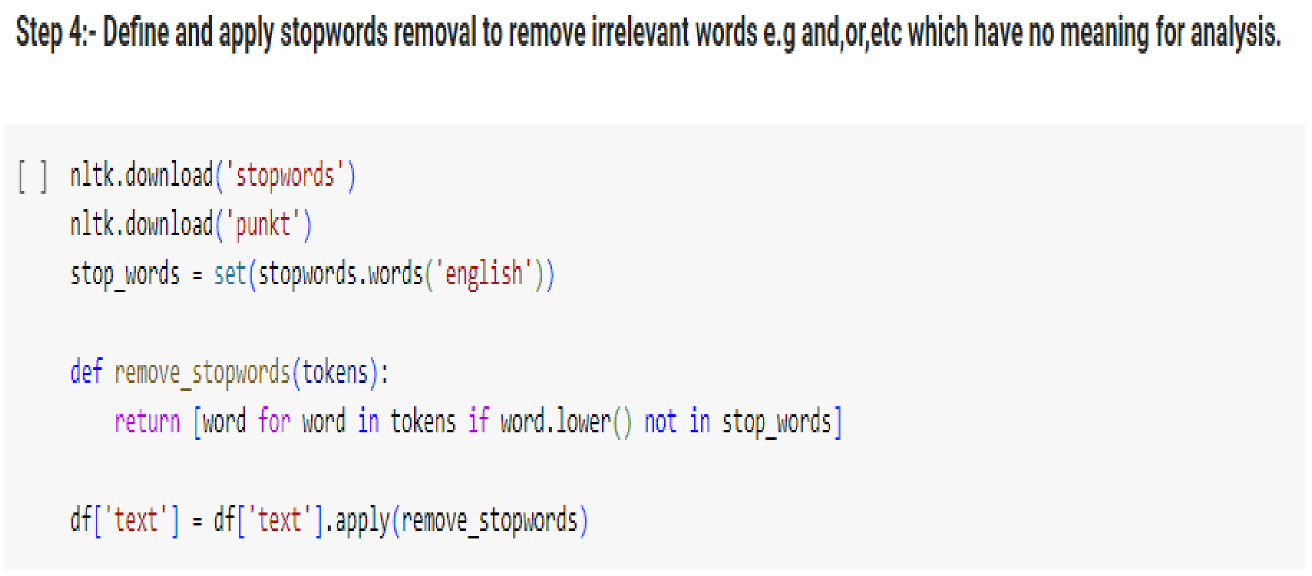
```
[ ]  df
```

```
           sentiment                                              text
      0            0      [is, upset, that, he, cant, upd, his, facebook...
      1            0        [kenich, i, div, many, tim, for, the, bal, man...
      2            0          [my, whol, body, feel, itchy, and, lik, it, on...
      3            0        [nationwideclass, no, it, not, behav, at, al, ...
      4            0                          [kweside, not, the, whol, crew]
    ...          ...                                                    ...
1048567            4         [my, grandm, is, mak, dinenr, with, my, mum]
1048568            4  [midmorn, snack, tim, a, bowl, of, chees, nood...
```

Figure 9 Data after stemming

IX. Stop Words: -

After the Stemming our next step is to remove the irrelevant words like “is”, “am” or “there” etc. That is achieve by using stop words module of nltk. That module has almost 120 above words which are part of English language in order to complete sentence but in NLP we do not need that words for our Model. It reduces the Complexity of our text and also reduce computational time and cost. The code to achieve that is shown in figure below,


Step 4:- Define and apply stopwords removal to remove irrelevant words e.g and,or,etc which have no meaning for analysis.

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

     def remove_stopwords(tokens):
         return [word for word in tokens if word.lower() not in stop_words]

     df['text'] = df['text'].apply(remove_stopwords)
```

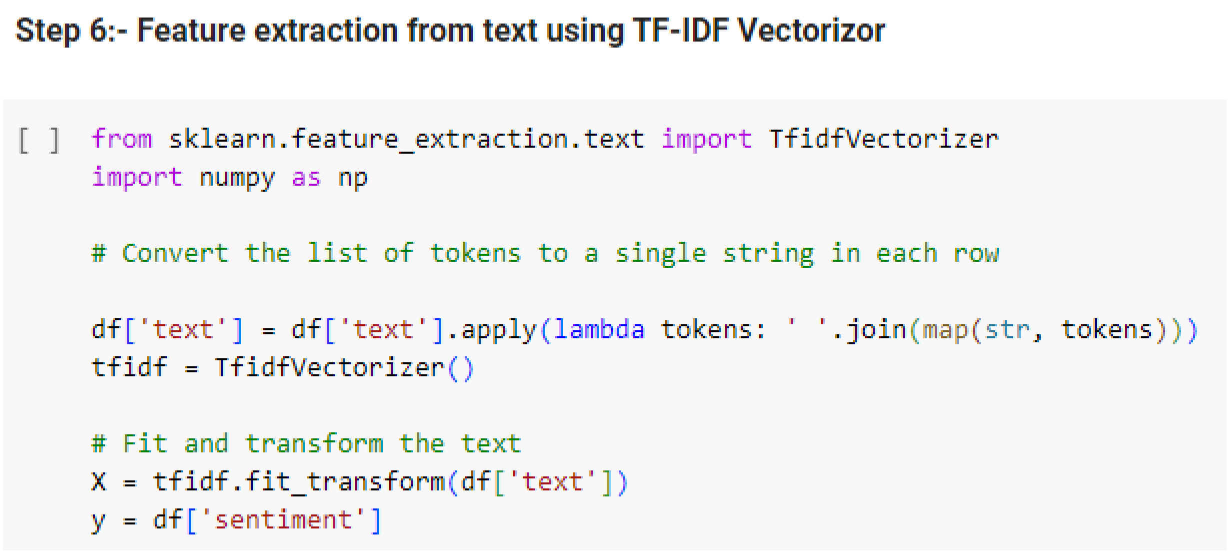
Figure 10 Applying Stop Words on Data

X. Feature Extraction through TF-IDF: -

Feature extraction is important part in NLP Projects and it involves extraction of the words which have meaning in the text. I use TF-IDF method which is Term frequency inverse Document Frequency. It is the model of Sklearn which performs following tasks,

- Convert text into Tokens
- Lower case the text and remove punctuations by default
- Calculate how many times a word occurs in document
- Find the Importance of that word in the Document
- Make a Vector based on above operations which have score of each word

The important point here is that we should do first 4 steps before feature extraction otherwise he counts "is", "am" or other words which are irrelevant which effects our accuracy score of model.



Step 6:- Feature extraction from text using TF-IDF Vectorizor

```
[ ]  from sklearn.feature_extraction.text import TfidfVectorizer
     import numpy as np

     # Convert the list of tokens to a single string in each row

     df['text'] = df['text'].apply(lambda tokens: ' '.join(map(str, tokens)))
     tfidf = TfidfVectorizer()

     # Fit and transform the text
     X = tfidf.fit_transform(df['text'])
     y = df['sentiment']
```

Figure 11 Feature Extraction from data

XI. Model Selection and Evaluation: -

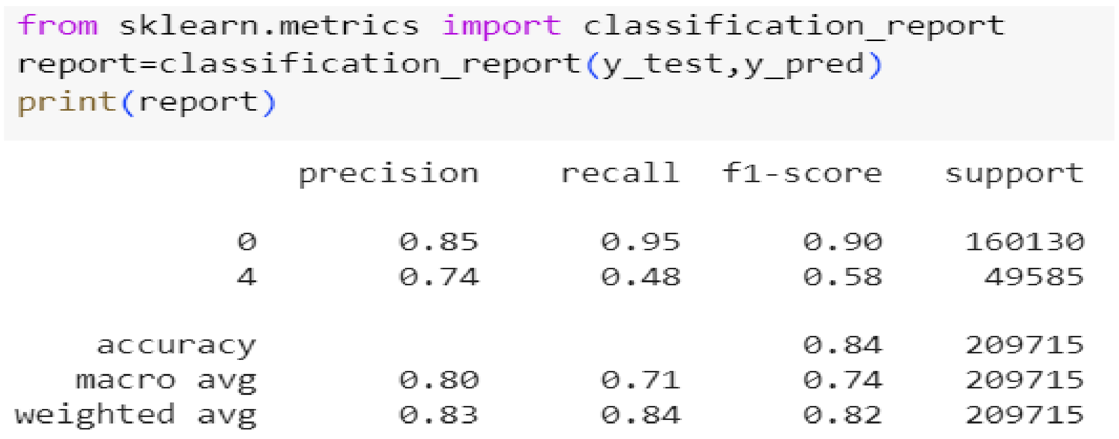
The Model selection is a difficult and technical part of any AI Project. Selection of right model is depending on the following things,

- Type of Problem (Regression, Classification, Clustering)
- Nature of data
- Data Size
- Data Complexity
- Computation Time and Cost
- Scalability

Many other factors are there but important once are shown above. As we have a Classification and Supervised Learning Problem so I select Logistic Regression for that as I have binary Classification. But we may use other ML or Deep Learning Models also. The hit or trial method is to evaluate 5 best ML models that you think and choose the best Model which perform well on that. It’s not a rule to just use Deep Learning Methods or Transformers for better results. Sometimes these models not perform well but Simple model perform better than these complex models.

“The model selection is totally depending on type and size of data and patterns inside data sometimes Complex models does not perform well for a given dataset and simple Logistic model works.” **Andrew Ng**

With the increase in Data set the computational Cost also increase specially for DLM and image processing tasks or NLP base tasks. I have not too big data so I prefer to use Logistic Regression Classifier because it take less time for training and results are quite good rather than SVC and DLM. It is the Mathematics which make things applicable as NLP is text base but we convert text into Numeric Vectors and now we can apply any classification model on that.



```python
from sklearn.metrics import classification_report
report=classification_report(y_test,y_pred)
print(report)
```

```
              precision    recall  f1-score   support

           0       0.85      0.95      0.90    160130
           4       0.74      0.48      0.58     49585

    accuracy                           0.84    209715
   macro avg       0.80      0.71      0.74    209715
weighted avg       0.83      0.84      0.82    209715
```

Figure 12 Results with Logistic Regression Classifier

The model accuracy is about 84% which is quite satisfactory and we can increase its accuracy further if we need. That accuracy is on the test Data set which we obtain by splitting.

XII. Challenges encountered: -

The big challenge is to Understand how NLP works and Preprocessing the data is quite difficult which takes a lot of time because I was not familiar with the NLP base tasks before its first time so quite difficult.