MINOR PROJECT (Group-17)

Context based Sentiment Analysis with Sarcasm Detection on tweeter tweets

Bachelors of Technology in Computer Science and Engineering

Submitted by

Amarjeet Chaudhary - 1806116

Swapna Sahani - 1806134

Rohit Singh - 1806005

Under the Supervision of

Dr. Anubha Maurya

NIT Patna

Department of Computer Science & Engineering

National Institute of Technology Patna

Patna-800005

JAN-MAY, 2021

राष्ट्रीय प्रौद्योगिकी संस्थान पटना
NATIONAL INSTITUTE OF TECHNOLOGY PATNA

ACKNOWLEDGEMENT

We would like to acknowledge and express our deepest gratitude to our mentor Dr. Anubha Maurya, Computer Science & Engineering Department, National Institute of Technology Patna for the valuable guidance, sympathy and co-operation for providing necessary facilities and sources during the entire period of this project. We would also like to convey our sincere gratitude to the Head of Department and all the faculties of Computer Science & Engineering Department who have enlightened us during our studies. The faculties and cooperation received from the technical staff of Department of Computer Science & Engineering is thankfully acknowledged.

1.Amarjeet Chaudhary 1806116

2.Swapna Sahani 1806134

3.Rohit Singh 1806005

Content

1. Introduction

1.1 Context

This project has been done as a part of our course for the B. Tech at NIT Patna of Computer Science and Technology. Supervised by Dr. Anubha Maurya , we had four months to fulfill the requirements in order to succeed the module. Every two weeks, a meeting was organized to show and report our progress and fix the next objectives.

1.2 Motivations

Being extremely interested in everything having a relation with the Machine Learning, the independent project was a great occasion to give us the time to learn and confirm our interest for this field. The fact that we can make estimations, predictions and give the ability for machines to learn by themselves is both powerful and limitless in term of application possibilities. We can use Machine Learning in Finance, Medicine, almost everywhere. That's why we decided to conduct my project around the Machine Learning.

This project was motivated by my desire to investigate all the complete description of the tweets in field of machine learning

1.3 Problem Statement

We are working onto the sentiment analysis on twitter including context classification and sarcasm detection. One of the most exciting topic presently where a tweet need to be checked in all the aspects and have a clear idea about it. So that we don't have a misinterpretation about it.

1.4 Problem Solution

Looking into the problem statement mentioned above we have come up with the solution for respective modules such as context classification using Topic Modeling, Sarcasm Detection using LSTM neural network module and Sentiment Analysis using Regression module. We applied the idea with tweets and try to figure out all the semantics aspects of the tweets.

1.5 Algorithm

```
Def comp(s, threshold, value):
    # if sarcastic_value > threshold return false to reverse the polarity else return true for no change
    if(value > threshold):
        return false
    else:
        return true
Def model(s1, s2):
```

```
data = [s1, s2]
# STEP 1  find the topic for the data
topic_decided = ContextClassificationModule(s1, s2)

# STEP 2 calculate the similarity between data
similarity_percentage = SentenceSimilarityModule(s1, s2, topic_decided)

# STEP 3 calculate the sarcastic percentage for each data
sarcasm_percentage = SarcasmModule(s1, s2)

# STEP 4 calculate sentiment using previous output as input along with comp() function for reverse the output where needed
final_output = SentimentAnalysisModule(s1, s2, sarcasm_percentage, comp)
```

NOTE: ContextClassificationModule() , SentenceSimilarityModule() , SarcasmModule() , SentimentAnalysisModule()

are the modules created and takes input as maintained

1.6 Workflow

A tweet can play an important role if its significance at the particular period is impressive so, before moving forward and further process or retweet we must be clear about its sarcastic behavior and its context after which we can check its sentiment so keeping this in mind we have also processed the tweets in the similar fashion.

a) We have checked whether the tweets is sarcastic or not.
b) Simultaneously we checked the context of the tweet.
c) Based on the output we have decided the sentiment of the tweet.

After processing the tweet is this manner we will be having a clear idea about the tweet and further retweet keeping all the aspects into consideration, which will really help in keeping the tweet concise and to the point.

Data Flow Diagram :

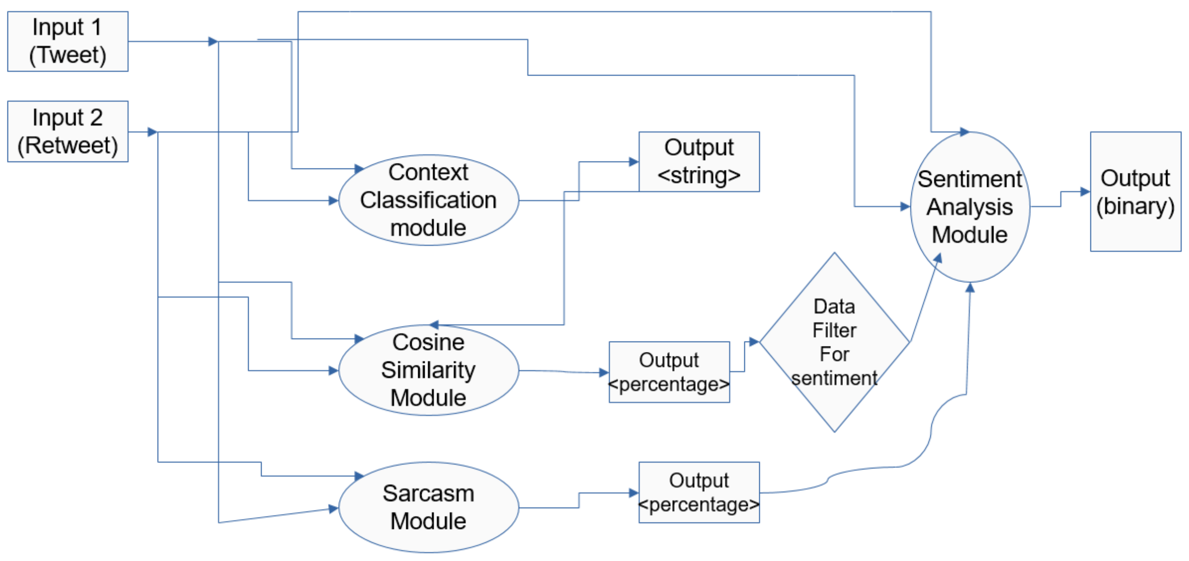


Fig: 1 - Data flow Diagram for the project

1.7 Modeling

Small fragments of words need to sum up to form a successful module thus following this we have come up with four module which will be combined together in a sequence to output a desire result.

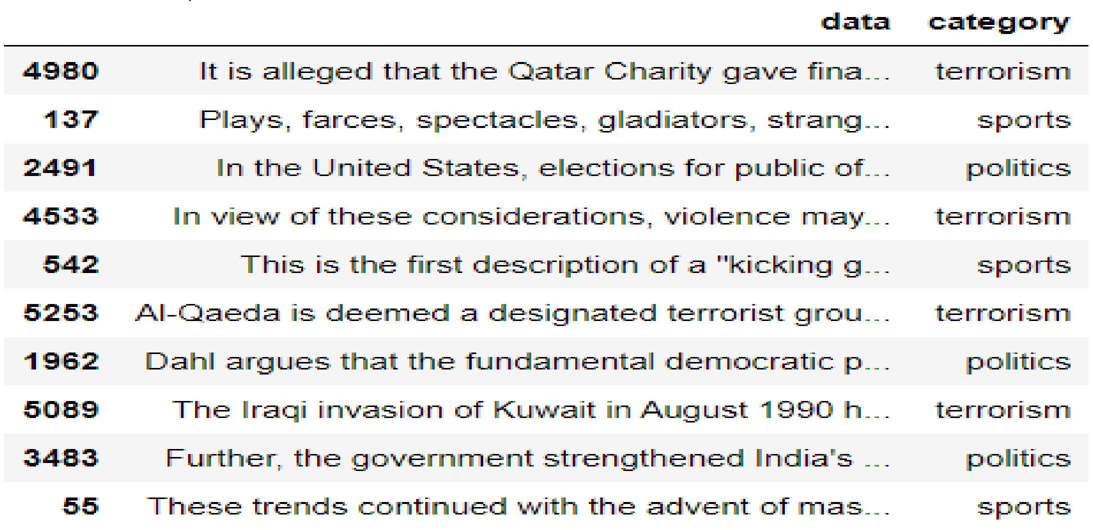
i. **Context classification Module –** It is used to determine context of tweets.
ii. **Sarcasm Detection Module –** It is used to determine the probabilistic value of tweet in context of sarcasm.
iii. **Sentiment Analysis Module –** It is used to do a binary classification based upon the polarity of the tweet.
iv. **Sentence Similarity Module –** It is used to determine the similarity between two tweets.

1.8 Dataset

- **Dataset-I for Context classification:**

  We have created a dataset of 5434 sentences from Wikipedia of different topics such as 'sports', 'politics' and 'terrorism'. There are approximate equal sentences of each topic (approximately

1800 from each)



	data	category
4980	It is alleged that the Qatar Charity gave fina...	terrorism
137	Plays, farces, spectacles, gladiators, strang...	sports
2491	In the United States, elections for public of...	politics
4533	In view of these considerations, violence may...	terrorism
542	This is the first description of a "kicking g...	sports
5253	Al-Qaeda is deemed a designated terrorist grou...	terrorism
1962	Dahl argues that the fundamental democratic p...	politics
5089	The Iraqi invasion of Kuwait in August 1990 h...	terrorism
3483	Further, the government strengthened India's ...	politics
55	These trends continued with the advent of mas...	sports

Table: 1.1 - Dataset-I for Context Classification

- **Dataset-II for Sarcasm Detection**:

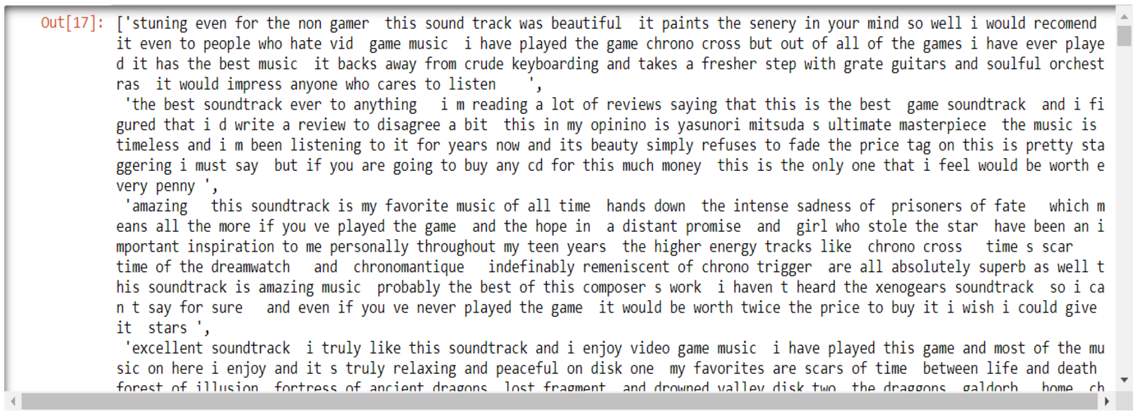
  We have a data set of 26709 tweets in English coming from Kaggle. It is composed of 2 columns that are headline and class. Class contains a binary value, 0 if the tweet is not sarcasm, 1 if the tweet is sarcasm

Headline	is_sarcastic
former versace store clerk sues over secret 'b...	0
the 'roseanne' revival catches up to our thorn...	0
mom starting to fear son's web series closest ...	1
boehner just wants wife to listen, not come up...	1
j.k. rowling wishes snape happy birthday in th...	0

Table: 1.2 - Dataset-II for Sarcasm Detection

- **Dataset-III for Sentiment Analysis:**
- We have a dataset of 20000 tweets in English coming from the bz format file where 9743 negative and 10257 positive. It is composed of 2 information first is the binary value according to the positive or negative. . We are only interested by the Sentiment text corresponding to our label class taking a binary value, 0 if the tweet is negative, 1 if the tweet is positive and the Sentiment

Text containing the tweets in a raw format.



```
Out[17]: ['stuning even for the non gamer  this sound track was beautiful  it paints the senery in your mind so well i would recomend
         it even to people who hate vid  game music  i have played the game chrono cross but out of all of the games i have ever playe
         d it has the best music  it backs away from crude keyboarding and takes a fresher step with grate guitars and soulful orchest
         ras  it would impress anyone who cares to listen    ',
          'the best soundtrack ever to anything   i m reading a lot of reviews saying that this is the best  game soundtrack  and i fi
         gured that i d write a review to disagree a bit  this in my opinino is yasunori mitsuda s ultimate masterpiece  the music is
         timeless and i m been listening to it for years now and its beauty simply refuses to fade the price tag on this is pretty sta
         ggering i must say  but if you are going to buy any cd for this much money  this is the only one that i feel would be worth e
         very penny ',
          'amazing   this soundtrack is my favorite music of all time  hands down  the intense sadness of  prisoners of fate   which m
         eans all the more if you ve played the game  and the hope in  a distant promise  and  girl who stole the star  have been an i
         mportant inspiration to me personally throughout my teen years  the higher energy tracks like  chrono cross   time s scar
         time of the dreamwatch   and  chronomantique   indefinably remeniscent of chrono trigger  are all absolutely superb as well t
         his soundtrack is amazing music  probably the best of this composer s work  i haven t heard the xenogears soundtrack  so i ca
         n t say for sure   and even if you ve never played the game  it would be worth twice the price to buy it i wish i could give
         it  stars ',
          'excellent soundtrack  i truly like this soundtrack and i enjoy video game music  i have played this game and most of the mu
         sic on here i enjoy and it s truly relaxing and peaceful on disk one  my favorites are scars of time  between life and death
         forest of illusion  fortress of ancient dragons  lost fragment  and drowned valley disk two  the draggons  galdorb   home  ch
```

▪ **Table: 1.3 - Dataset-III for Sentiment Analysis**

2. Proposed Solution

Sentiment analysis, also refers as opinion mining, is a sub machine learning task where we want to determine which is the general sentiment of a given document including the topic it belong to and the tweet is sarcasm or not. Using machine learning techniques and natural language processing we can extract the subjective information of a document and try to classify it according to its polarity such as positive or negative. It is a really useful analysis since we could possibly determine the overall opinion about a selling objects, or predict stock markets for a given company like, if most people think positive about it, possibly its stock markets will increase, and so on. Sentiment analysis is actually far from to be solved since the language is very complex (objectivity/subjectivity, negation, vocabulary, grammar) but it is also why it is very interesting to working on. In this project we choose to try to classify tweets from Twitter into "positive" or "negative" sentiment by building a module based on probabilities. Twitter is a microblogging website where people can share their feelings quickly and spontaneously by sending a tweets limited by 140 characters. You can directly address a tweet to someone by adding the target sign "@" or participate to a topic by adding an hastag "#" to your tweet. Because of the usage of Twitter, it is a perfect source of data to determine the current overall opinion about anything.

2.1 Context classification

2.1.1 Description

A **topic model** is a type of statistic model for discovering the abstract "topics" that occur in a collection of documents. Topic modeling is a frequently used text-mining tool for discovery of hidden semantic structures in a text body. The "topics" produced by topic modeling techniques are clusters of similar words.

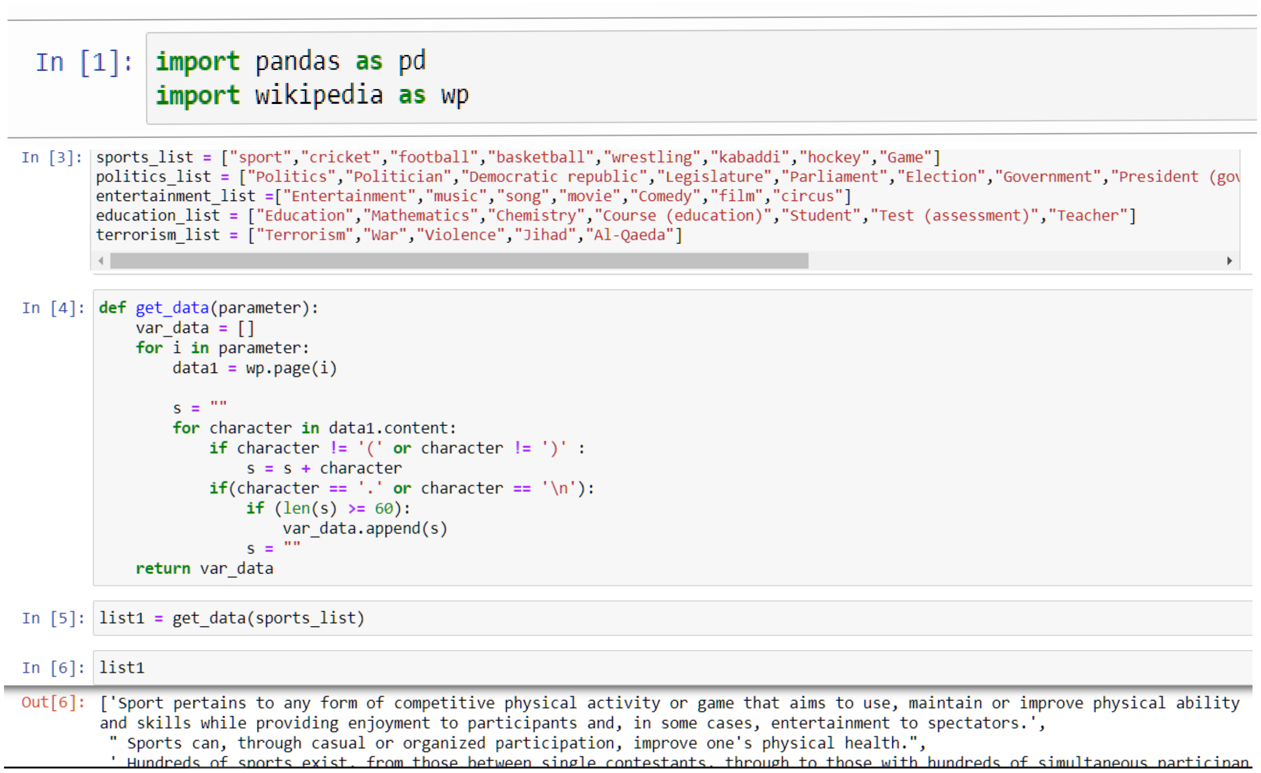
In this method we choose to try to classify tweets into different topics such as "Sports", "Politics" and "Terrorism" by building a module using an unsupervised learning technique.

2.1.2 Data

To gather the data many options are possible but to find a way of getting a corpus of tweets of different topics, we need to take of having a balanced data set, meaning we should have an approximately equal number of tweets from each topics, but it needs also to be large enough. Indeed, more the data we have, more we can train our classifier and more the accuracy will be.

After many researches, when we didn’t found any categorized dataset of such topics. Then we create our own dataset from extracting tweets from Wikipedia of each topics. Our dataset is composed of 2 columns that are ‘data’ and ‘category’. We are only interested by ‘data’ column, which we used for deciding topics of each tweet.

Dataset – I mentioned is our context Classification dataset, extracted from Wikipedia



```
In [1]: import pandas as pd
        import wikipedia as wp
```

```
In [3]: sports_list = ["sport","cricket","football","basketball","wrestling","kabaddi","hockey","Game"]
        politics_list = ["Politics","Politician","Democratic republic","Legislature","Parliament","Election","Government","President (gov
        entertainment_list =["Entertainment","music","song","movie","Comedy","film","circus"]
        education_list = ["Education","Mathematics","Chemistry","Course (education)","Student","Test (assessment)","Teacher"]
        terrorism_list = ["Terrorism","War","Violence","Jihad","Al-Qaeda"]
```

```
In [4]: def get_data(parameter):
            var_data = []
            for i in parameter:
                data1 = wp.page(i)

                s = ""
                for character in data1.content:
                    if character != '(' or character != ')' :
                        s = s + character
                    if(character == '.' or character == '\n'):
                        if (len(s) >= 60):
                            var_data.append(s)
                        s = ""
            return var_data
```

```
In [5]: list1 = get_data(sports_list)
```

```
In [6]: list1
```

```
Out[6]: ['Sport pertains to any form of competitive physical activity or game that aims to use, maintain or improve physical ability
        and skills while providing enjoyment to participants and, in some cases, entertainment to spectators.',
         " Sports can, through casual or organized participation, improve one's physical health.",
         ' Hundreds of sports exist, from those between single contestants, through to those with hundreds of simultaneous participan
```

Fig 2.1.A - Data generation for Context classification from Wikipedia

In the above Table 1.1 showing the first ten tweet of different topic decided we can already notice some particularities and difficulties that are going to encounter during the pre-processing steps.

We can visualize a bit more the dataset by making a chart of how many tweets each topics does contains.

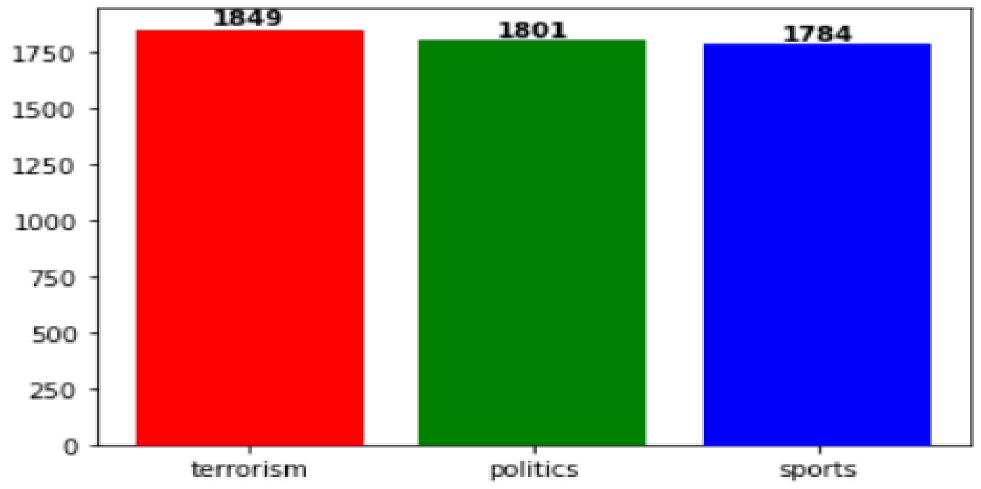


Fig: 2.1.B – histogram representation of dataset-I

We have exactly 1849 tweets of terrorism, 1801 tweets of politics and 1784 tweets of sports that the dataset is well-balanced.

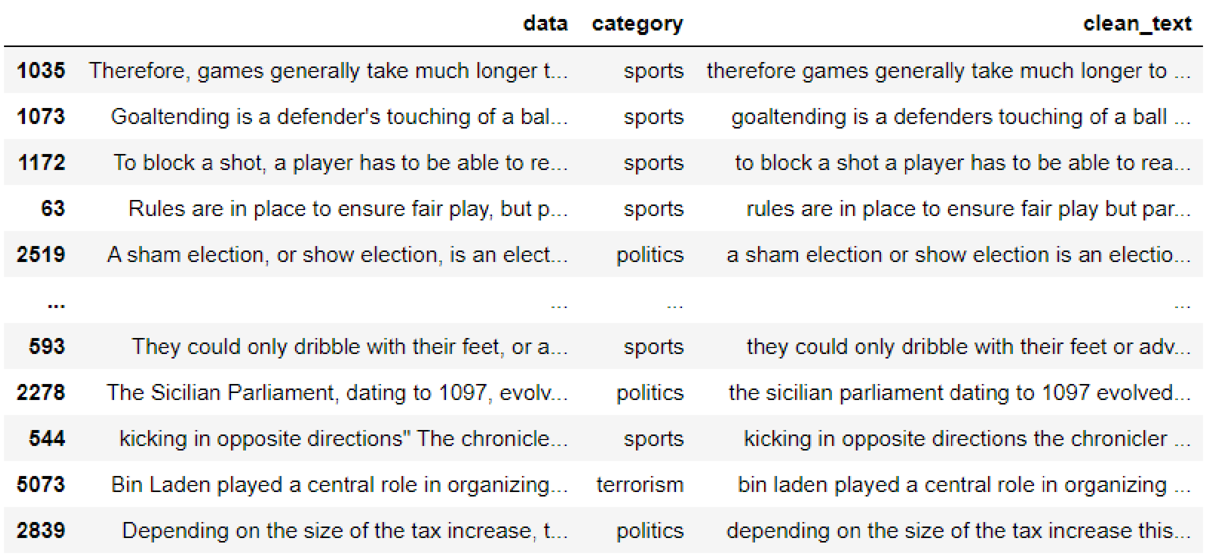
2.1.3 Pre-processing

Now we will pre-process out tweets. It is a very important since all the modifications that we are going to during this process will directly impact the classifier's performance. The pre-processing includes cleaning, normalization, transformation, feature extraction and selection, etc. The result of pre-processing will be consistent and uniform data that are workable to maximize the classifier's performance.

All of the tweets are pre-processed by passing through the following steps in the same order.

Lowering and cleaning text-

We are creating a function, which will take the input tweet and change the all alphabets of our tweet to LOWER_CASE and remove special characters such as ([ ), (,), (.), ("), (\ ), ('), (!), (@), (#), ($), (%), (^), (&), (*), (( ), ( )), ({), (}), (?), (/), (;), (`), (~), (:), (<), (>), (+), (=), (-) and ( ]). Finally returns a tweet which will help to obtain maximum accuracy.

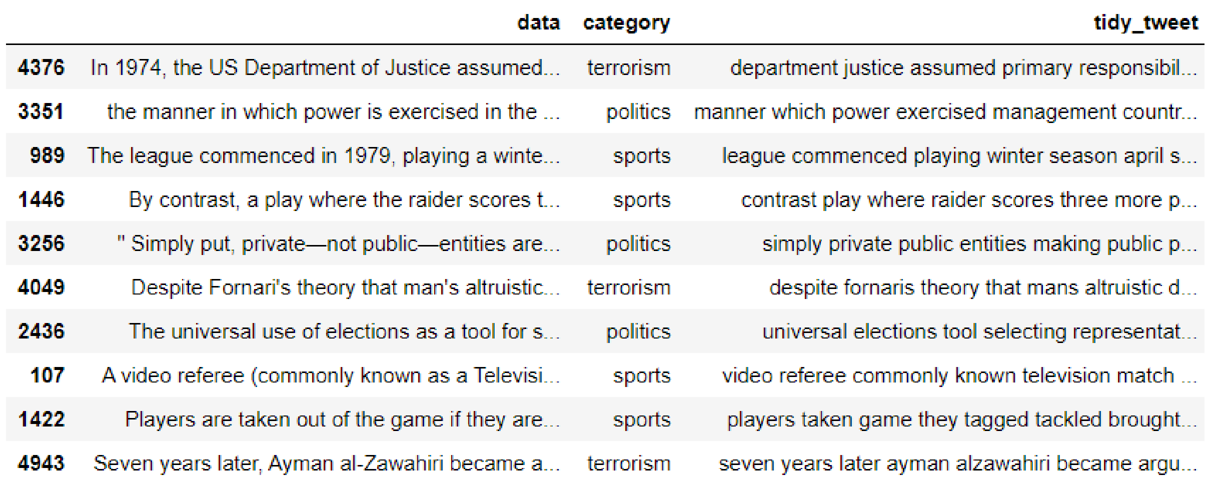


	data	category	clean_text
1035	Therefore, games generally take much longer t...	sports	therefore games generally take much longer to ...
1073	Goaltending is a defender's touching of a bal...	sports	goaltending is a defenders touching of a ball ...
1172	To block a shot, a player has to be able to re...	sports	to block a shot a player has to be able to rea...
63	Rules are in place to ensure fair play, but p...	sports	rules are in place to ensure fair play but par...
2519	A sham election, or show election, is an elect...	politics	a sham election or show election is an electio...
...	...	...	...
593	They could only dribble with their feet, or a...	sports	they could only dribble with their feet or adv...
2278	The Sicilian Parliament, dating to 1097, evolv...	politics	the sicilian parliament dating to 1097 evolved...
544	kicking in opposite directions" The chronicle...	sports	kicking in opposite directions the chronicler ...
5073	Bin Laden played a central role in organizing...	terrorism	bin laden played a central role in organizing ...
2839	Depending on the size of the tax increase, t...	politics	depending on the size of the tax increase this...

Table: 2.1.A – lowercase cleaned tweets

Here, the column 'clean-text' store all the cleaned texts returned by the function.

Further we continue the pre-processing and removing all unwanted stuffs except alphabets. We are also removing word whose length is less than four. After this processing most of the word such as in, are, is, ok, to etc. will remove which have no use for deciding the topics.



	data	category	tidy_tweet
4376	In 1974, the US Department of Justice assumed...	terrorism	department justice assumed primary responsibil...
3351	the manner in which power is exercised in the ...	politics	manner which power exercised management countr...
989	The league commenced in 1979, playing a winte...	sports	league commenced playing winter season april s...
1446	By contrast, a play where the raider scores t...	sports	contrast play where raider scores three more p...
3256	" Simply put, private—not public—entities are...	politics	simply private public entities making public p...
4049	Despite Fornari's theory that man's altruistic...	terrorism	despite fornaris theory that mans altruistic d...
2436	The universal use of elections as a tool for s...	politics	universal elections tool selecting representat...
107	A video referee (commonly known as a Televisi...	sports	video referee commonly known television match ...
1422	Players are taken out of the game if they are...	sports	players taken game they tagged tackled brought...
4943	Seven years later, Ayman al-Zawahiri became a...	terrorism	seven years later ayman alzawahiri became argu...

Table: 2.1.B – dataset-I after removing stop words

The above table has a column named as 'tidy -tweet' is the final text column after pre-processing. It doesn't contain any stop words, numeric value, special character, etc.

NULL value and Blanks

Now we will work on the dataset to check if there is any Null value or blank space at any row then we will simply remove those row to get a better train module.

2.1.4 NLP

2.1.4.i TF-IDF

TF-IDF is an abbreviation for Term Frequency Inverse Document Frequency. This is very common algorithm to transform text into a meaningful representation of numbers which is used to fit machine algorithm for prediction. After transformation It gives a sparse matrix of N x M, where N is the number of rows and M is the number different words.

In **TfidfVectorizer** we consider **overall document weightage** of a word. It helps us in dealing with most frequent words. Using it we can penalize them. TfidfVectorizer weights the word counts by a measure of how often they appear in the documents.

In our vectorizer we are passing three parameters max_df, min_df and stop_words.

Max_df is used for removing terms that appear **too frequently. In our module** we passed max_df value equals to 0.9, which means ignore terms that appear in **more than 90% of the documents.**

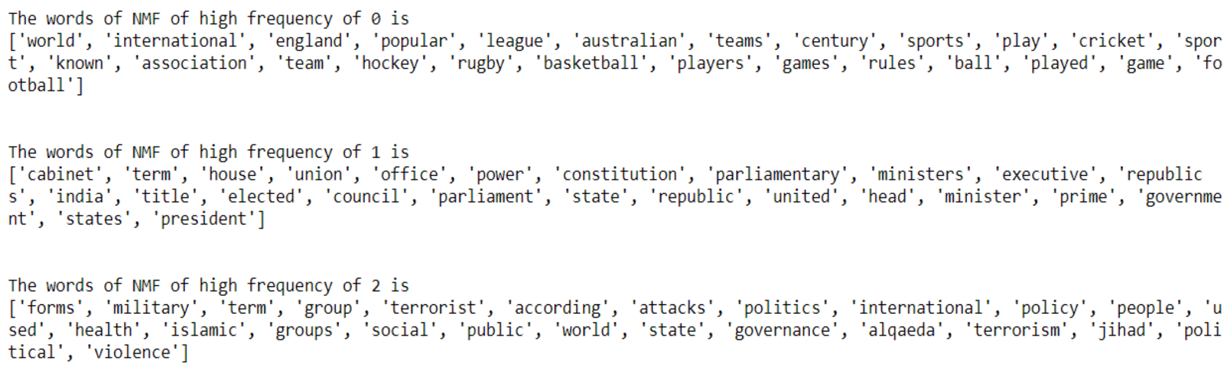
Min_df is used for removing terms that appear **too infrequently. In our module** we passed min_df value equals to 2, which means ignore terms that appear in **less than 5 tweets.**

Stop_words are those words which we don't want the vectorizer to consider. In our project we passed the english stop words to not considerable by the tf-idf vectorizer.

2.1.4.ii NMF

Non-Negative Matrix Factorization (NMF) is an **unsupervised technique** use for labeling of topics. The way it works is that, NMF decomposes (or factorizes) high-dimensional vectors into a lower-dimensional representation. These lower-dimensional vectors are **non-negative** which also means their coefficients are non-negative. The high-dimensional vectors are going to be tf-idf weights but it can be really anything including word vectors or a simple raw count of the words.

NMF module categorizes the document into 'n' components and return an array of words for each component. The array are sorted from lower frequency words to higher frequency words. With the help of these words we need to decide the topic of each component. In our project we took 'n' value as three because we want to categorize our document in three topics.

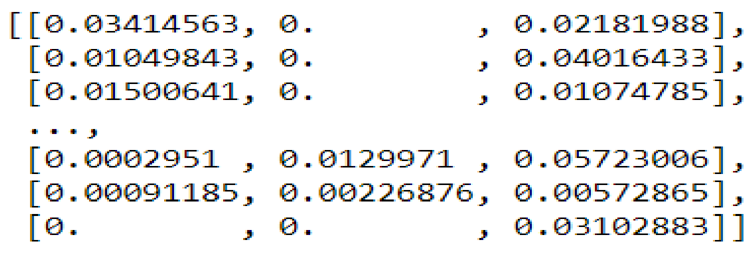


```
The words of NMF of high frequency of 0 is
['world', 'international', 'england', 'popular', 'league', 'australian', 'teams', 'century', 'sports', 'play', 'cricket', 'spor
t', 'known', 'association', 'team', 'hockey', 'rugby', 'basketball', 'players', 'games', 'rules', 'ball', 'played', 'game', 'fo
otball']


The words of NMF of high frequency of 1 is
['cabinet', 'term', 'house', 'union', 'office', 'power', 'constitution', 'parliamentary', 'ministers', 'executive', 'republic
s', 'india', 'title', 'elected', 'council', 'parliament', 'state', 'republic', 'united', 'head', 'minister', 'prime', 'governme
nt', 'states', 'president']


The words of NMF of high frequency of 2 is
['forms', 'military', 'term', 'group', 'terrorist', 'according', 'attacks', 'politics', 'international', 'policy', 'people', 'u
sed', 'health', 'islamic', 'groups', 'social', 'public', 'world', 'state', 'governance', 'alqaeda', 'terrorism', 'jihad', 'poli
tical', 'violence']
```

Fig: 2.1.C - 25 high frequency words of each component

Above image shows the 25 high frequency words of each component. By looking at the word belonging to the component we have chosen our first component as 'sports', second components as 'politics' and the third and last component as 'terrorism'.

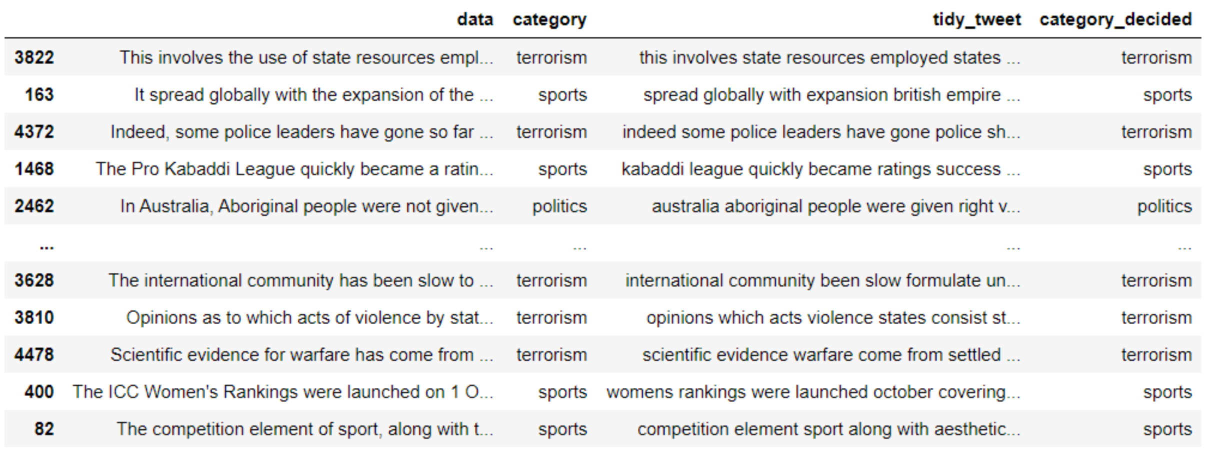
After Transforming the module a sparse matrix generated by tf-idf vectorizer, it return a probability array for each tweet. The probability array contain n probability values for each tweets. The probability values of probability array of each tweet decides in which component the tweet belongs.



```
[[0.03414563, 0.        , 0.02181988],
 [0.01049843, 0.        , 0.04016433],
 [0.01500641, 0.        , 0.01074785],
 ...,
 [0.0002951 , 0.0129971 , 0.05723006],
 [0.00091185, 0.00226876, 0.00572865],
 [0.        , 0.        , 0.03102883]]
```

Fig: 2.1.D - probability array of each tweets

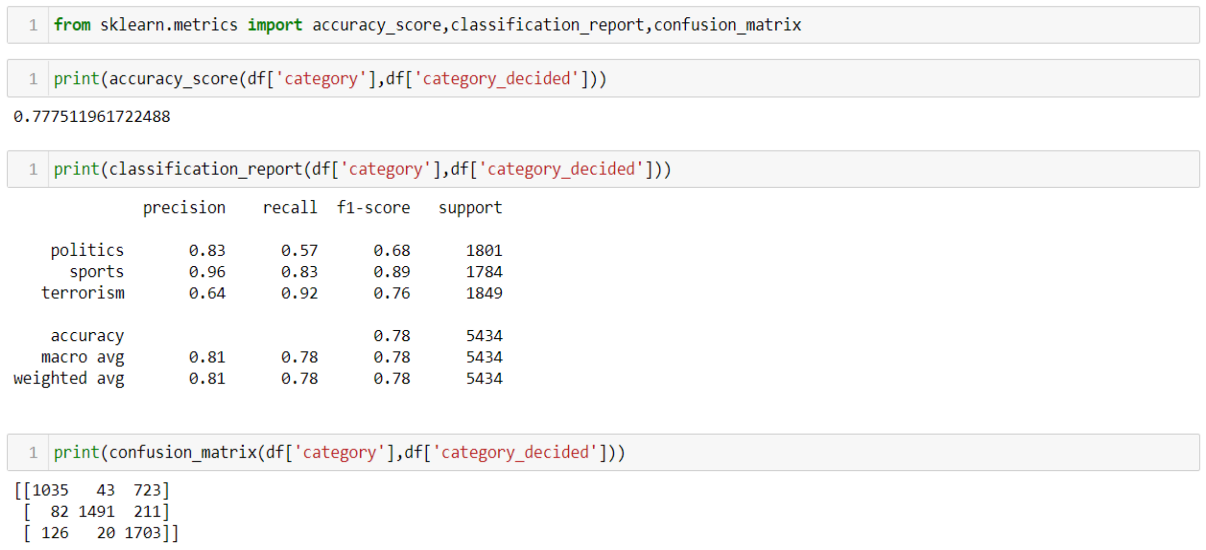
Here is the probability array of our module having three component and probability value of tweet belonging to each component. Highest probability value is the topic in which the tweet belong.



	data	category	tidy_tweet	category_decided
3822	This involves the use of state resources empl...	terrorism	this involves state resources employed states ...	terrorism
163	It spread globally with the expansion of the ...	sports	spread globally with expansion british empire ...	sports
4372	Indeed, some police leaders have gone so far ...	terrorism	indeed some police leaders have gone police sh...	terrorism
1468	The Pro Kabaddi League quickly became a ratin...	sports	kabaddi league quickly became ratings success ...	sports
2462	In Australia, Aboriginal people were not given...	politics	australia aboriginal people were given right v...	politics
...	...	...	...	...
3628	The international community has been slow to ...	terrorism	international community been slow formulate un...	terrorism
3810	Opinions as to which acts of violence by stat...	terrorism	opinions which acts violence states consist st...	terrorism
4478	Scientific evidence for warfare has come from ...	terrorism	scientific evidence warfare come from settled ...	terrorism
400	The ICC Women's Rankings were launched on 1 O...	sports	womens rankings were launched october covering...	sports
82	The competition element of sport, along with t...	sports	competition element sport along with aesthetic...	sports

Table: 2.1.C - The above Table show the category decided by the module.

Accuracy_score:0.777



```
from sklearn.metrics import accuracy_score,classification_report,confusion_matrix
```

```
print(accuracy_score(df['category'],df['category_decided']))
```

```
0.777511961722488
```

```
print(classification_report(df['category'],df['category_decided']))
```

```
              precision    recall  f1-score   support

    politics       0.83      0.57      0.68      1801
      sports       0.96      0.83      0.89      1784
   terrorism       0.64      0.92      0.76      1849

    accuracy                           0.78      5434
   macro avg       0.81      0.78      0.78      5434
weighted avg       0.81      0.78      0.78      5434
```

```
print(confusion_matrix(df['category'],df['category_decided']))
```

```
[[1035   43  723]
 [  82 1491  211]
 [ 126   20 1703]]
```

Fig: 2.1.E - accuracy, confusion matrix and classification report for context classification module.

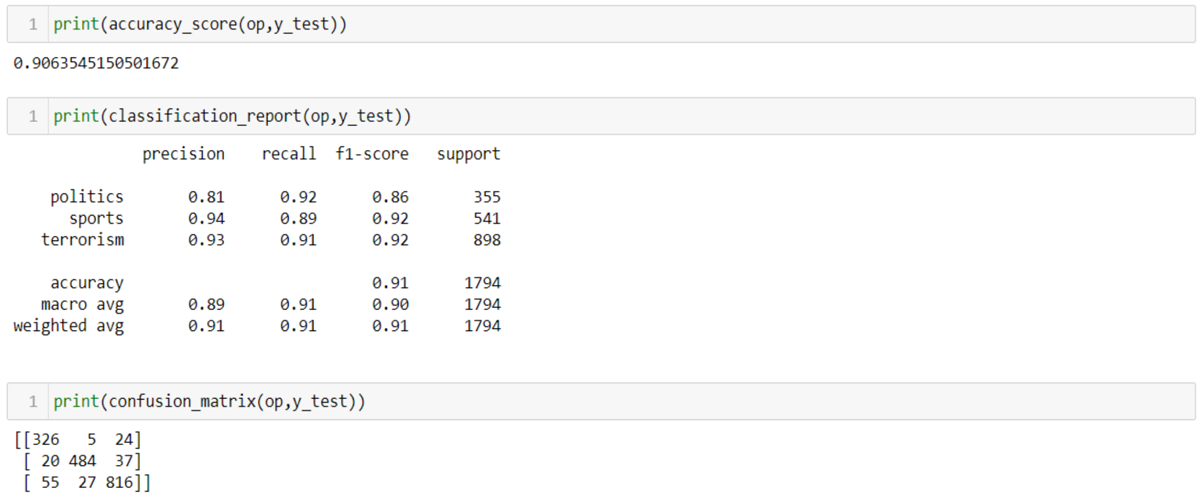
Once we have decide the categories, we can now focus on the machine learning part. We have used **Linear SVM** to train a module on our decided categories.

2.1.4.iii <u>Linear SVM :</u>

Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as **Linear SVM** classifier.

The SVM module works on numbers not on strings, So we again need to use Tf-idf Vectorizer.

Accuracy_score:0.906



```
print(accuracy_score(op,y_test))
```

```
0.9063545150501672
```

```
print(classification_report(op,y_test))
```

```
              precision    recall  f1-score   support

    politics       0.81      0.92      0.86       355
      sports       0.94      0.89      0.92       541
   terrorism       0.93      0.91      0.92       898

    accuracy                           0.91      1794
   macro avg       0.89      0.91      0.90      1794
weighted avg       0.91      0.91      0.91      1794
```

```
print(confusion_matrix(op,y_test))
```

```
[[326   5  24]
 [ 20 484  37]
 [ 55  27 816]]
```

Fig: 2.1.F - accuracy, confusion matrix and classification report for trained SVM module.

2.2 Sarcasm Detection

2.2.1 Description

Sarcasm detection is a very narrow research field in NLP, a specific case of sentiment analysis where instead of detecting a sentiment in the whole spectrum, the focus is on **sarcasm**. Therefore the task of this field is to detect if a given text is **sarcastic** or not.

The first problem we come across is that, unlike in sentiment analysis where the sentiment categories are very clearly defined (love objectively has a positive sentiment, hate a negative sentiment no matter who you ask or what language you speak), the borders of sarcasm aren't that well defined. And it is crucial that before starting to detect it, **to have a notion of what sarcasm is.** Sarcasm is the use of language that normally signifies the opposite in order to mock or convey contempt.

2.2.2 Data

To gather the data many options are possible. We have built a program to collect automatically a corpus of tweets based on two classes, either a sarcasm or not.

Additionally to find a way of getting a corpus of tweets, we need to take of having a balanced data set, meaning we should have a similar numbers of both the classes, but it needs also to be large enough. Indeed, more the data we have, more we can train our neural network and more the accuracy will be.

After many researches, we found dataset-II.

2.2.3 Preprocessing:

Removing all the emoticons and doing all the required preprocessing we obtain the cleaned data.

	headline	is_sarcastic	clean_text
0	former versace store clerk sues over secret 'b...	0	former versace store clerk sues over secret bl...
1	the 'roseanne' revival catches up to our thorn...	0	the roseanne revival catches up to our thorny ...
2	mom starting to fear son's web series closest ...	1	mom starting to fear sons web series closest t...
3	boehner just wants wife to listen, not come up...	1	boehner just wants wife to listen not come up ...
4	j.k. rowling wishes snape happy birthday in th...	0	jk rowling wishes snape happy birthday in the ...

Table: 2.2.A - Data after preprocessing

2.2.4 Machine Learning

2.2.4.i LSTM

Long Short Term Memory networks – usually just called "LSTMs" – are a special kind of RNN, capable of learning long-term dependencies. LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn.

All recurrent neural networks have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer.

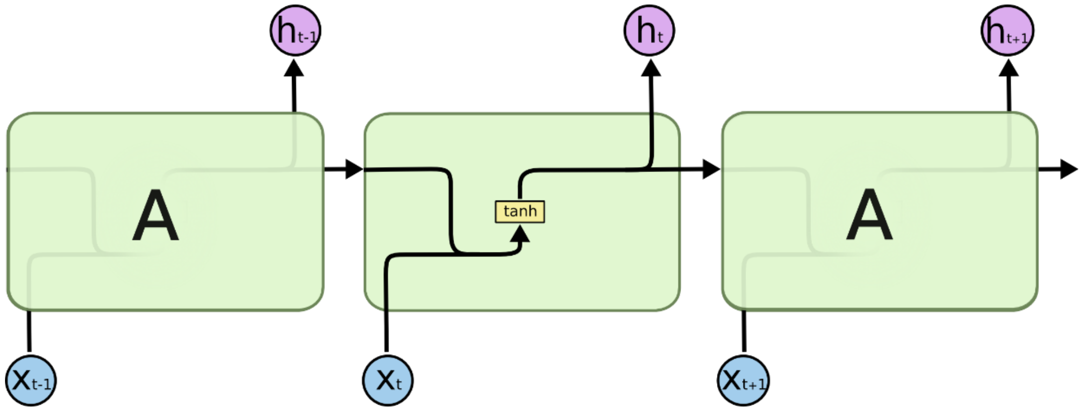


Fig: 2.2.A - The repeating module in a standard RNN contains a single layer.

LSTMs also have this chain like structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four, interacting in a very special way.

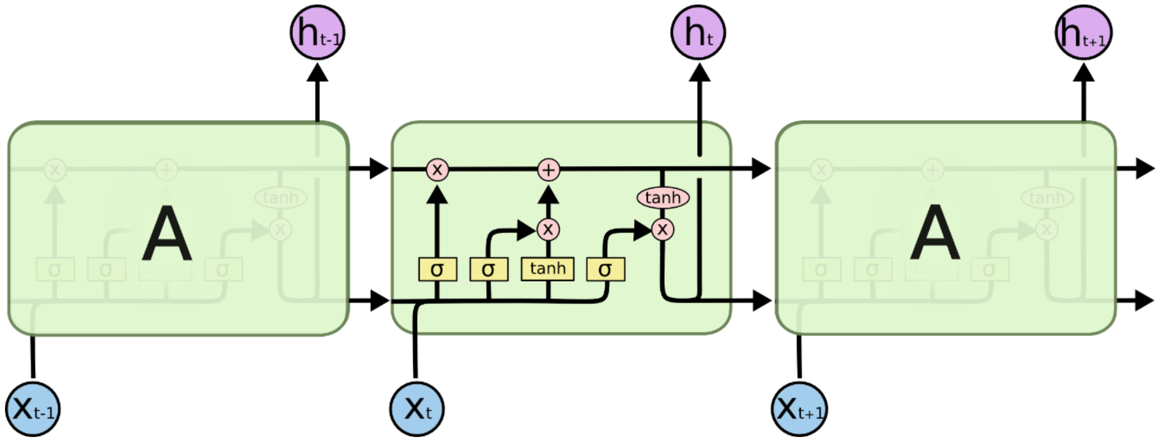


Fig: 2.2.B -The repeating module in an LSTM contains four interacting layers.

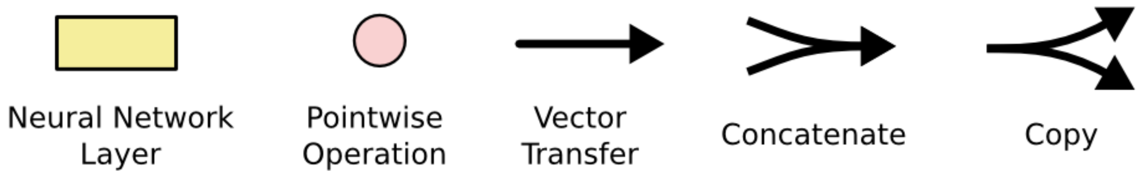


Fig: 2.2.C –symbolic representation

Don't worry about the details of what's going on. We'll walk through the LSTM diagram step by step later. For now, let's just try to get comfortable with the notation we'll be using.

In the above diagram, each line carries an entire vector, from the output of one node to the inputs of others. The pink circles represent point-wise operations, like vector addition, while the yellow boxes are learned neural network layers. Lines merging denote concatenation, while a line forking denote its content being copied and the copies going to different locations.

2.2.4.ii Output

1s - loss: 0.0359 - accuracy: 0.9880 - val_loss: 1.3236 - val_accuracy: 0.8364

A good validation accuracy we have obtained using LSTM module.

Sarcasm which plays a vital role in any conversation need to be handled with care because it's misinterpretation will lead to a conflicting outcome. And can lead the chain of tweets to deviate into other direction leaving the main topic.

After this we would classify the tweets into different limited domains.

2.3 Sentiment Analysis

2.3.1 Description

Sentiment analysis, also refers as opinion mining, is a sub machine learning task where we want to determine which is the general sentiment of a given document. Using machine learning techniques and natural language processing we can extract the subjective information of a document and try to classify it according to its polarity such as positive or negative. It is a really useful analysis since we could possibly determine the overall opinion about a selling objects, or predict stock markets for a given company like, if most people think positive about it, possibly its stock markets will increase, and so on. Sentiment analysis is actually far from to be solved since the language is very complex (objectivity/subjectivity, negation, vocabulary, grammar) but it is also why it is very interesting to working on.

2.3.2 Data

To gather the data many options are possible. In some previous paper researches, they built a program to collect automatically a corpus of tweets based on two classes, “positive” and “negative”, by querying Twitter with two type of sentiment : Others make their own dataset of tweets my collecting and annotating them manually which very long and fastidious. Additionally to find a way of getting a corpus of tweets, we need to take of having a balanced data set, meaning we should have an equal number of positive and negative tweets, but it needs also to be large enough. Indeed, more the data we have, more we can train our classifier and more the accuracy will be.

After many attempt, we found a dataset-III

2.3.3 Pre-processing

Now that we have the corpus of tweets and all the resources that could be useful, we can preprocess the tweets. It is a very important since all the modifications that we are going to during this process will directly impact the classifier’s performance. The preprocessing includes cleaning, normalization, transformation, feature extraction and selection, etc. The result of preprocessing will be consistent and uniform data that are workable to maximize the classifier's performance.

2.3.4 Machine Learning

Once we have applied the different steps of the preprocessing part, we can now focus on the machine learning part. There is the modules we used in sentiment analysis to classify a tweet into positive or negative: Logistic Regression which gave us the good accuracy without over fitting. Logistic Regression is known to be the module giving the best results but in this project. After trying several module and keeping both the accuracy and over-fitting into consideration we finally declared this module to be quite in the better place.

2.3.4.i Logistic Regression

Logistic regression is a supervised **learning** classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes. ... Mathematically, a **logistic regression** module predicts P(Y=1) as a function of X. It is one of the simplest ML algorithms that can be used for various classification problems such as spam detection, Diabetes prediction, cancer detection etc.

Types of Logistic Regression

Generally, logistic regression means binary logistic regression having binary target variables, but there can be two more categories of target variables that can be predicted by it. Based on those number of categories, Logistic regression can be divided into following types −

Binary or Binomial

In such a kind of classification, a dependent variable will have only two possible types either1 and 0. For example, these variables may represent success or failure, yes or no, win or loss etc.

Multinomial

In such a kind of classification, dependent variable can have 3 or more possible ***unordered*** types or the types having no quantitative significance. For example, these variables may represent "Type A" or "Type B" or "Type C".

Ordinal

In such a kind of classification, dependent variable can have 3 or more possible ***ordered*** types or the types having a quantitative significance. For example, these variables may represent "poor" or "good", "very good", "Excellent" and each category can have the scores like 0,1,2,3.

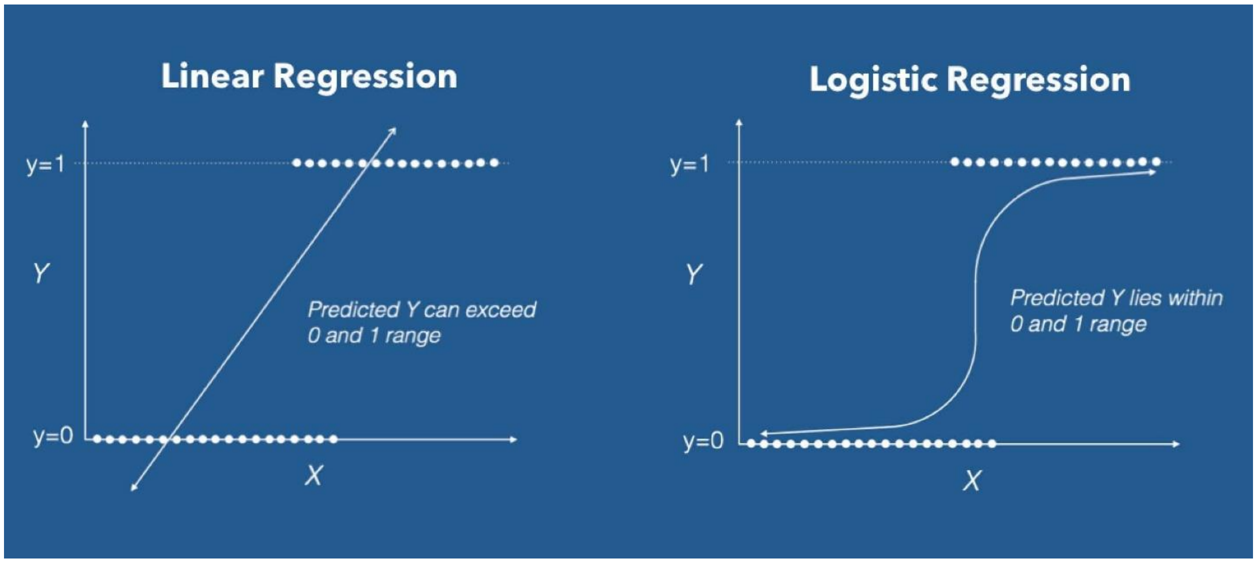

Linear Regression
y=1
Y
Predicted Y can exceed 0 and 1 range
y=0
X
Logistic Regression
y=1
Y
Predicted Y lies within 0 and 1 range
y=0
X

Fig: 2.3.A – Linear and Logistic Regression

Here we can see that the Logistic Regression Module give a probabilistic value between 1 and 0.

2.3.4.ii Logistic Function (Sigmoid Function):

The sigmoid function is a mathematical function used to map the predicted values to probabilities.

It maps any real value into another value within a range of 0 and 1.

The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form. The S-form curve is called the Sigmoid function or the logistic function.

In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.

2.3.4.iii Logistic Regression Equation:

The Logistic regression equation can be obtained from the Linear Regression equation. The mathematical steps to get Logistic Regression equations are given below:

We know the equation of the straight line can be written as:

$$\boldsymbol{y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n} \qquad \textbf{(1)}$$

In Logistic Regression y can be between 0 and 1 only, so for this let's divide the above equation by (1-y):

$$\boldsymbol{\frac{y}{1-y}; 0 for y = 0, \wedge \infty for y = 1} \qquad \textbf{(2)}$$

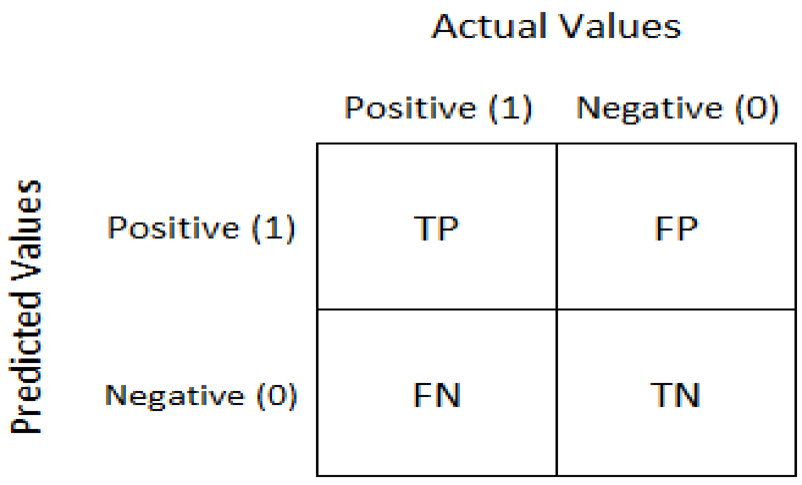
But we need range between -[infinity] to +[infinity], then take logarithm of the equation it will become:

$$\boldsymbol{log\left[\frac{y}{1-y}\right] = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n} \qquad \textbf{(3)}$$

The above equation is the final equation for Logistic Regression.

Test Accuracy of the result

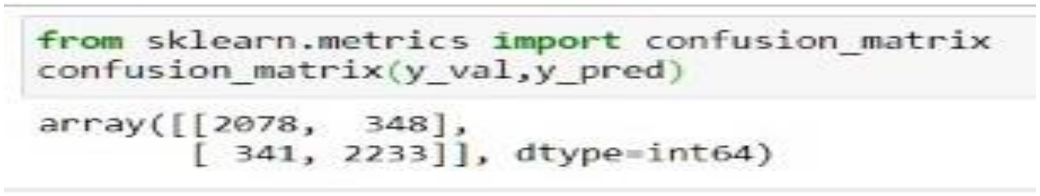
Confusion Matrix



Predicted Values		Actual Values: Positive (1)	Actual Values: Negative (0)
	Positive (1)	TP	FP
	Negative (0)	FN	TN

Table: 2.3.A –Confusion Matrix

The confusion matrix here to check the accuracy of the classification. To create it, we need to import the confusion matrix function of the sklearn library. After importing the function, we will call. The function takes two parameters, mainly **y_val**(the actual values) and **y_pred** (the targeted value return by the classifier). Below is the confusion matrix of our module:



```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_val,y_pred)
```

```
array([[2078,  348],
       [ 341, 2233]], dtype=int64)
```

Fig: 2.3.B – confusion matrix for sentiment analysis module

Looking into the confusion matrix we can see that the dataset used for the testing purpose which is the 25% of the overall dataset.

Notice that to evaluate our classifier we two methods, the F1 score and a confusion matrix. The F1 Score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. It a measure of a classifier's accuracy.

Where the precision is the number of true positives (the number of items correctly labeled as belonging to the positive class) divided by the total number of elements labeled as belonging to the positive class.

The recall is the number of true positives divided by the total number of elements that actually belong to the positive class.

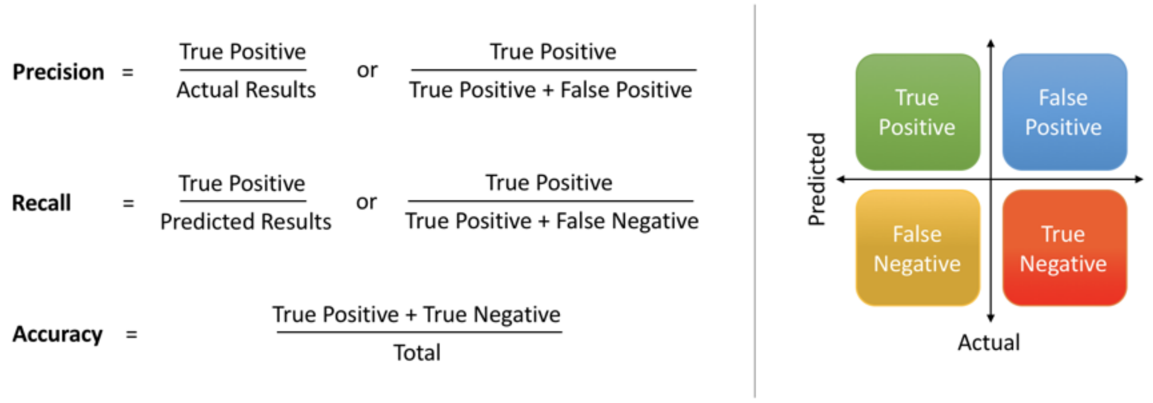
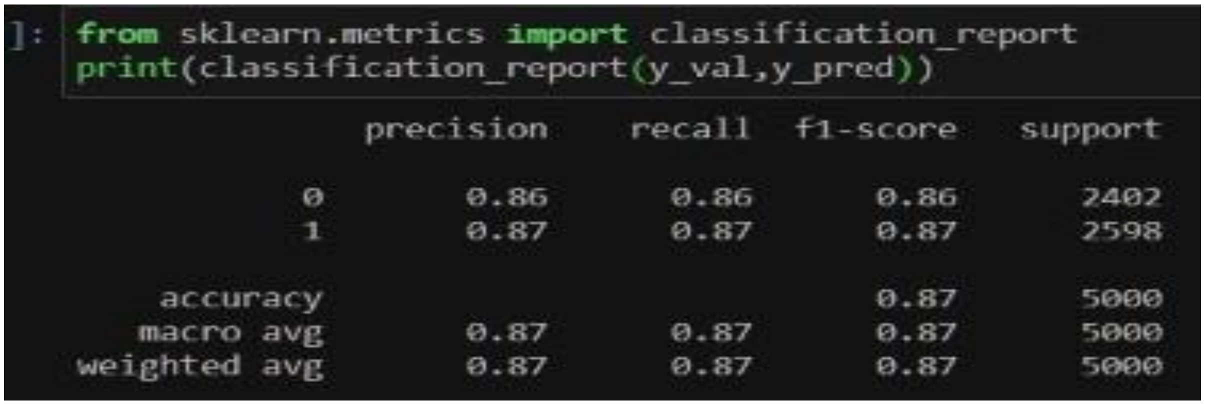

Precision = True Positive / Actual Results or True Positive / True Positive + False Positive
Recall = True Positive / Predicted Results or True Positive / True Positive + False Negative
Accuracy = True Positive + True Negative / Total
True Positive
False Positive
False Negative
True Negative
Predicted
Actual

Fig: 2.3.C –Precision, Recall and Accuracy

There is a tradeoff between precision and recall where increasing one decrease the other and we usually use measures that combine precision and recall such as F measure.

At last we can show the classification report of our module.



```
]: from sklearn.metrics import classification_report
   print(classification_report(y_val,y_pred))

                 precision    recall  f1-score   support

              0       0.86      0.86      0.86      2402
              1       0.87      0.87      0.87      2598

       accuracy                           0.87      5000
      macro avg       0.87      0.87      0.87      5000
   weighted avg       0.87      0.87      0.87      5000
```

Fig: 2.3.D – classification report for sentiment analysis module

2.4 Sentence Similarity

Sentence similarity measures the similarity between two vectors of an inner product space. It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction. It is often used to measure document similarity in text analysis .It is one of the metric to measure the text-similarity between two documents irrespective of their size in **Natural language Processing**. If the Cosine Similarity score is 1, it means two vectors have the same orientation. The value closer to 0 indicates that the two documents have less similarity.

A document can be represented by thousands of attributes, each recording the frequency of a particular word (such as a keyword) or phrase in the document. Thus, each document is an object represented by what is called a **term-frequency vector**. For example, the table below, we see that Document contains five instances of the word team, while hockey occurs three times. The word coach is absent from the entire document, as indicated by a count value of 0. Such data can be highly asymmetric.

Document	***team***	***coach***	***hockey***	***baseball***	***soccer***	***penalty***	***score***	***win***	***loss***	***season***
Document1	5	0	3	0	2	0	0	2	0	0
Document2	3	0	2	0	1	1	0	1	0	1
Document3	0	7	0	2	1	0	0	3	0	0
Document4	0	1	0	0	1	2	2	0	3	0

Table: 2.4 – document for sentence similarity

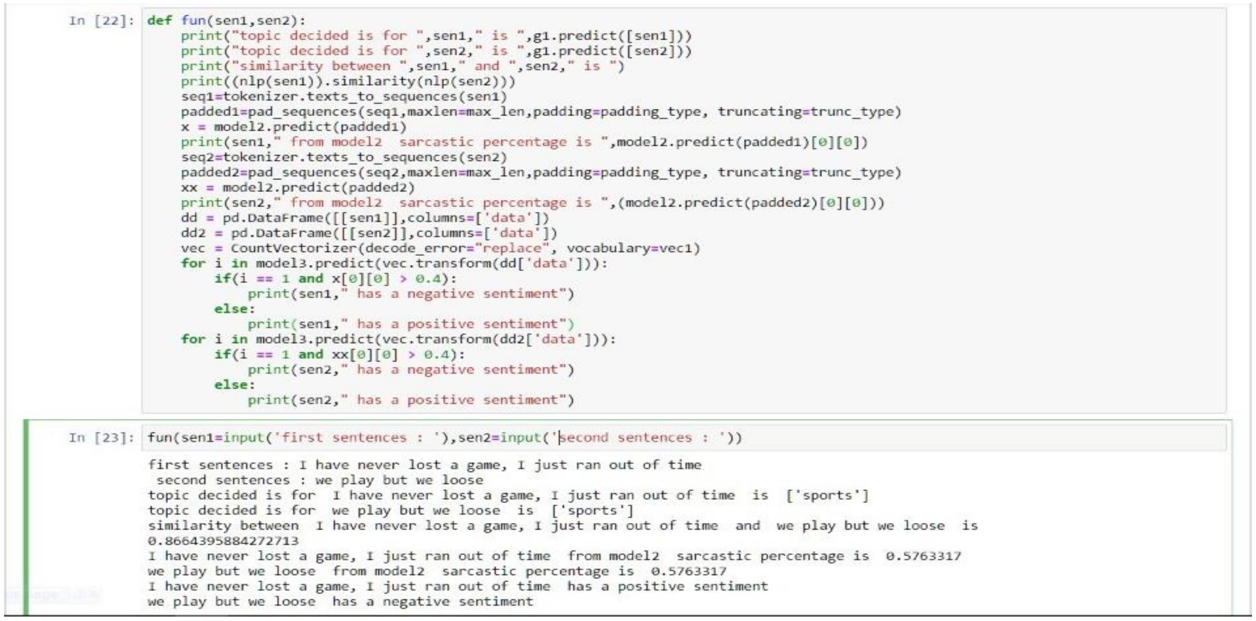
- Cos(d1,q) = vector_multiplication(d1,q) / (|d1|*|q|)
- Let’s → d1 = <a1, a2, a3> and q =<b1, b2, b3>
- Cosine(d1,q) = ((a1b1+a1b2+…...+a3b3)/(sqrt(a1a1+a2a2+a3a3)sqrt(b1b1+b2b2+b3b3))
- Similarly we can calculate for Cosine(d2,q)
- Note : Cosine(a,b) = Cosine(b,a) , Cosine(theta) = Cosine(-theta)

- Dis1 = cosine(d1,q)
- Dis2 = cosine(d2,q)
- Compare Dis1 and Dis2, smaller will be nearer to the query vector.

$$\boldsymbol{similarity = cos(\theta) = \frac{A \cdot B}{||A||||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} {A_i}^2}\sqrt{\sum_{i=1}^{n} {B_i}^2}}} \qquad (4)$$

2.5 Combined model

After summing up all the modules we have come up with a model which will be provided with two tweets as an input strings as shown in the dataflow diagram, which will be classify the context of the tweets, if the tweets are out of context then it will not be further processed otherwise after this the same tweets will be passed through sarcasm module to determine the degree of sarcasm further we will do a binary classification using Sentiment analysis module into positive and negative tweets, at last we will determine the similarity between the tweets.



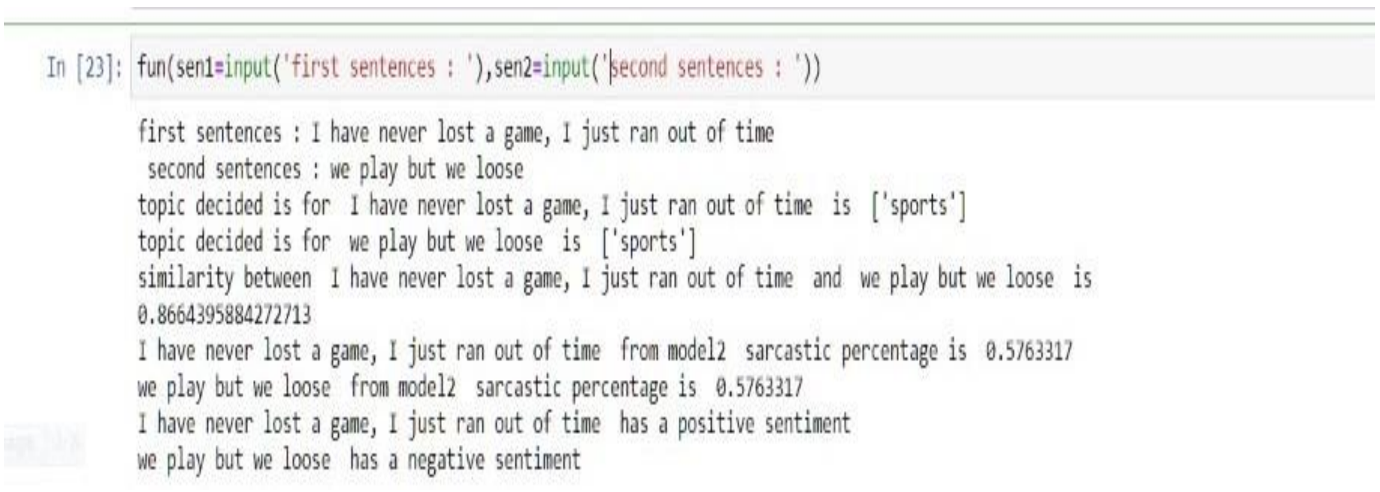
```
In [22]: def fun(sen1,sen2):
             print("topic decided is for ",sen1," is ",g1.predict([sen1]))
             print("topic decided is for ",sen2," is ",g1.predict([sen2]))
             print("similarity between ",sen1," and ",sen2," is ")
             print((nlp(sen1)).similarity(nlp(sen2)))
             seq1=tokenizer.texts_to_sequences(sen1)
             padded1=pad_sequences(seq1,maxlen=max_len,padding=padding_type, truncating=trunc_type)
             x = model2.predict(padded1)
             print(sen1," from model2  sarcastic percentage is ",model2.predict(padded1)[0][0])
             seq2=tokenizer.texts_to_sequences(sen2)
             padded2=pad_sequences(seq2,maxlen=max_len,padding=padding_type, truncating=trunc_type)
             xx = model2.predict(padded2)
             print(sen2," from model2  sarcastic percentage is ",(model2.predict(padded2)[0][0]))
             dd = pd.DataFrame([[sen1]],columns=['data'])
             dd2 = pd.DataFrame([[sen2]],columns=['data'])
             vec = CountVectorizer(decode_error="replace", vocabulary=vec1)
             for i in model3.predict(vec.transform(dd['data'])):
                 if(i == 1 and x[0][0] > 0.4):
                     print(sen1," has a negative sentiment")
                 else:
                     print(sen1," has a positive sentiment")
             for i in model3.predict(vec.transform(dd2['data'])):
                 if(i == 1 and xx[0][0] > 0.4):
                     print(sen2," has a negative sentiment")
                 else:
                     print(sen2," has a positive sentiment")

In [23]: fun(sen1=input('first sentences : '),sen2=input('second sentences : '))

         first sentences : I have never lost a game, I just ran out of time
          second sentences : we play but we loose
         topic decided is for  I have never lost a game, I just ran out of time  is  ['sports']
         topic decided is for  we play but we loose  is  ['sports']
         similarity between  I have never lost a game, I just ran out of time  and  we play but we loose  is
         0.8664395884272713
         I have never lost a game, I just ran out of time  from model2  sarcastic percentage is  0.5763317
         we play but we loose  from model2  sarcastic percentage is  0.5763317
         I have never lost a game, I just ran out of time  has a positive sentiment
         we play but we loose  has a negative sentiment
```

Fig: 2.5 - Combined model and output

3. Results

Hence now it's time to test our model providing it input and checking its context, sarcastic behavior and its sentiment.

Feeding few tweet to the model we will check its output:



```
In [23]: fun(sen1=input('first sentences : '),sen2=input('second sentences : '))

first sentences : I have never lost a game, I just ran out of time
 second sentences : we play but we loose
topic decided is for  I have never lost a game, I just ran out of time  is  ['sports']
topic decided is for  we play but we loose  is  ['sports']
similarity between  I have never lost a game, I just ran out of time  and  we play but we loose  is
0.8664395884272713
I have never lost a game, I just ran out of time  from model2  sarcastic percentage is  0.5763317
we play but we loose  from model2  sarcastic percentage is  0.5763317
I have never lost a game, I just ran out of time  has a positive sentiment
we play but we loose  has a negative sentiment
```

Fig: 3 – output of two tweets.

This is how our model will predict each and every aspects of a tweet.

4. Conclusion

Nowadays, sentiment analysis or opinion mining is a hot topic in machine learning. We are still far to detect the sentiments of a corpus of texts very accurately because of the complexity in the English language. In this project we tried to show the basic way of classifying tweets into positive or negative category using Logistic Regression as baseline and produce better results.

We have also worked on sarcasm and context deciding which will help for the retweet to be classified into its appropriate tweets and can be used at the public pages of ads and chat to not mess up and can clearly filter out the useful tweets.

5. References

[1] Arbel, N. (2018, Dec 21). *How LSTM networks solve the problem of vanishing gradients*. Retrieved from medium: https://medium.datadriveninvestor.com/how-do-lstm-networks-solve-the-problem-of-vanishing-gradients-a6784971a577

[2] Borcan, M. (2020, Jun 8). *TF-IDF Explained And Python Sklearn Implementation*. Retrieved from towardsdatascience: https://towardsdatascience.com/tf-idf-explained-and-python-sklearn-implementation-b020c5e83275

[3] Narkhede, S. (2018, May 17). *Understanding Logistic Regression*. Retrieved from towardsdatascience: https://towardsdatascience.com/understanding-logistic-regression-9b02c2aec102

[4] Pant, A. (2019, Jan 22). *Introduction to Logistic Regression*. Retrieved from towardsdatascience: https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148

[5] Saha, S. (2018, Dec 15). *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*. Retrieved from towardsdatascience: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

[6] Salgado, R. (2020, Apr 16). *Topic Modeling Articles with NMF*. Retrieved from medium: https://medium.com/@robert.salgado

[7] sawant, m. (2019, Jul 28). *Text Sentiments Classification with CNN and LSTM*. Retrieved from medium: https://medium.com/@mrunal68/text-sentiments-classification-with-cnn-and-lstm-f92652bc29fd

[8] *sklearn.decomposition.NMF*. (2007). Retrieved from scikit-learn: https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.NMF.html

[9] *sklearn.feature_extraction.text.TfidfVectorizer*. (2007). Retrieved from scikit-learn: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html