CHAPTER 1

INTRODUCTION

1.1 SENTIMENT ANALYSIS

In the early 2000s the article, **Thumbs up - Sentiment Classification using Machine Learning Techniques** published by **Bo Pang** and **Lillian lee** is considered to be the first article based on Sentiment analysis. Sentiment is nothing but the opinion expressed by the people. Analysis means examining something in detail. Thus, the Sentiment analysis means, examining the sentiment or view or opinion and finding something meaningful out of it. Sentiment analysis also referred to as opinion mining, helps to identify the emotions in the text. In other words, Sentiment analysis or opinion mining is the process to detect the opinion or emotion or view of the person through text, reviews, ratings, speeches, tweets etc… This analysis part is really important among the business people now-a-days. It helps them to understand the customer's opinion about their brand and helps them to change or modify the things that doesn't satisfy the customer and also to improvise their brands. This chapter deals with the importance of Sentiment Analysis with few applications and also the types, approaches and the challenges people face while analysing the sentiment of the text.

1.1.1 IMPORTANCE OF SENTIMENT ANALYSIS

People express their views or opinion about the product/event in many ways. One among them is expressing their opinion through text, in particular, it will be an unstructured text. The text holds lots of emotions, like happiness, anger, sadness, joy, etc. In order to find out something meaningful in the text, Sentiment analysis is carried out. One can manually find out the sentiment in a text only if the dataset is minimum, if not this analysis plays an important role. Also analysing the people's opinion is important because, it helps in finding out many useful things. For example, analysing the reviews of the product gives us better insights about the views of people about the product, which helps the business people to make some conclusions and also to make some changes in the product if needed. This is also important in promoting the brand and finding out the happy customers. Looking in a broader view, this analysis also helps to change the state of the company or growth of any organization etc. Some benefits of Sentiment analysis include, analysing the sentiment of the unstructured text over a large scale, providing the required thing to the desired one in the right time, predicting some future results, improvising the things in order to gain more attention and response etc.

1.1.2 APPLICATION OF SENTIMENT ANALYSIS

Sentiments are very important to organizations and government to make decision because they always want to find consumer or public opinions about their services or products. Many applications are there in analysing the sentiment of the text. Some real time applications are listed below:

- **Analysing the Product Reviews**

  In this type of analysis, the review of the particular product is taken and then the analysis is done. From this analysis, the opinion of the customer about the product/brand can be determined which is helpful in improvising the company.

- **Analysing the Social Media Content**

  The social media content i.e., data taken from various platforms like, Twitter, Facebook and so on, helps in determining the immediate responses of the people about the current trending activities like sports, elections, movies and any kind of decision policy made by the government.

- **Feedback in the Organization**

  The feedback from the people in any kind of organization/institution helps the particular community to make changes in the field which helps in either retaining the people in the field or to bring up an environment that in turn paves the way for the development of the organization.

1.1.3 SENTIMENT ANALYSIS CHALLENGES

Analysing the sentiment of the text is the hardest part of natural language processing. Despite the research developing in analysing the sentiment, people find it difficult in analysing the sentiment of the text accurately. This difficulty arises because the machines have to be trained to analyse and understand the human emotions. Some common challenges of machine-based sentiment analysis are given below:

- **Tone**

Nowadays, it is difficult to understand the tone of a person in a verbal communication. It's obvious that it will be even more difficult to understand the tone of the person in a non-verbal communication. When it comes to a dataset, which contains a lot of subjective and objective opinions, people find it difficult to analyse the sentiment accurately. A separate Tone detection algorithm may be used to overcome this challenge.

- **Context**

Context is nothing but the situation in which the words are said. Since it is impossible for the machines to understand the context in which the words are uttered, this also causes a change in the polarity of the whole sentence. That is, the words with less polarity may be ignored by the machines which provide a result which may change from the actual result.

- **Irony & Sarcasm**

In recent times, people use positive words to express their negativity towards the problem. This is termed as Irony. Sarcasm also means in a similar way. Even though the way the person speaks shows that he is joking, it is actually the opposite of what he says. This causes a change in polarity of the sentence. That is, the sentence takes the polarity value as positive which is supposed to be negative.

- **Comparison**

The unstructured data used in sentiment analysis also contains text which are comparative. It is difficult to detect the polarity of these comparative sentences because the text will only be a comparison of certain things and there will not be any words which could help to detect the polarity.

- **Emoji**

Using emoji in a text makes the text to be more expressive and makes the point clear to the receiver. That is why most of the social media-based content is overflowing with emoji. But the problem is that the NLP tasks are trained to be language specific. So, during the text mining process the NLP processor treats the emoji as a special character and removes it which in turn causes a change in the insight of the data.

1.1.4 DATASET

A dataset is a collection of related information. The information may belong to a particular area. Here we have taken the sports data from Twitter to analyse sentiment using a programming language. Twitter is a wide area to analyse because it consists of a large amount of data from public. Twitter data are used for different purposes like online surveys, review for products or services, crowdsourcing, etc…

1.1.5 PYTHON

The Sentiment analysis is a task that can be implemented using programming languages and also by the analysing tools. Python, R, SQL, Java, Scala, C/C++, etc... are the programming languages used for analysing the sentiment. Python is one of the most commonly used open-source programming languages to analyse sentiment. It is widely used programming language by the data scientists. Python was designed by **Guido van Rossum** and released in 1991. It has strong libraries like NumPy, SciPy, etc… for the mathematical functions.

1.2 PRELIMINARIES

In this section we will intend to recollect some of the prerequisites required for our study.

1.2.1 TYPES OF SENTIMENT ANALYSIS

Sentiment Analysis mainly focuses on Polarity, Emotions, Urgency, and Intentions. Also, it can be used for many purposes. Depending upon the purpose of the Sentiment analysis, it is classified into four types.

1. Fine- Grained Sentiment Analysis
2. Emotion Detection
3. Aspect based Sentiment Analysis
4. Intent Analysis

Fine-Grained Sentiment Analysis

The Fine-Grained Sentiment Analysis is used to understand the sentiment of the text in terms of polarity. This helps to understand the customer's feedback precisely. The Polarity of the text is classified into three categories namely Positive, Negative and Neutral. In some cases, the polarity is further classified into five categories i.e., Very Positive, Positive, Negative, Very Negative and Neutral depending upon the use case (for e.g., five-star ratings).

Emotion Detection

As the name says, this type of sentiment analysis is used to detect the emotions behind the text. It is used to detect emotions such as happiness, anger, frustration, worry, fear, panic etc. They use Lexicons (A list of words from a particular subject topic. Here it refers to a list of words which are used to express emotions) and Machine learning algorithms to detect the emotions. The drawback of this analysis is that people sometimes use different words to express their emotion, which in turn reverses the polarity of the sentence.

Aspect based Sentiment Analysis

In Aspect-based Sentiment analysis despite the polarity, the aspect about which the customer speaks or in general what sort of thing is highlighted is taken into account and improvising is made if they found any fault over there. This type of Sentiment analysis is very useful among business people to analyze, the products in particular and make some improvements or modify the features. This serves a great purpose in finding out the opinion of the customer.

Intent-based Sentiment Analysis

Intent refers to the purpose in which a particular thing is carried out. This Intent analysis is all about analyzing the key actions in the text so that a proper intention can be known. This analysis helps to find the people's intention (interest/ disinterest) accurately which helps in the betterment of the organization. When certain customers are disinterested about their products through this Intent analysis, it is easily known and advertisements or suggestions based on their interest will be shown.

1.2.2 SENTIMENT ANALYSIS APPROACHES

There are two ways in Sentiment analysis to find out the opinion of the text namely, Semantic Orientation Approach and the Machine Learning Approach. Let's look into both the approaches in detail.

1. Machine Learning Approach
2. Semantic Approach

1.2.3 MACHINE LEARNING APPROACH

Machine Learning treats Sentiment analysis as the text classification problem. This algorithm works based on the two methods, Supervised and Unsupervised learning methods. In Unsupervised Machine learning, proper inputs will not be given. The main target of the problem is identified by grouping the contents available in the data set whereas, in Supervised machine learning, the analysis is carried out by using a labelled dataset. These labelled datasets are used in the training process for making meaningful and required decisions.

1.2.4 MACHINE LEARNING ALGORITHMS

- **Naïve Bayes algorithm**: Naive bayes is a Supervised machine learning algorithm used to classify data.
- **Support Vector Machine (SVM):** SVM is also a supervised machine learning algorithm which is used for linear classification as well as regression.
- **Random Forest algorithm:** Random Forest is a supervised machine learning algorithm which is also used for classification. This algorithm is a collection of many decision trees.

1.2.5 SEMANTIC APPROACH

There are two different types of technique in Semantic approach, namely (i) Corpus-based and (ii) Dictionary/Lexicon/Knowledge-based approaches. The Corpus-based approach is the simplest one. A large dataset is required to detect the polarity of the text in this approach. The Knowledge-based approach uses pre-developed lexicons like SenticNet, WordNet, SenticWordNet, etc... to detect the polarity.

1.2.6 PYTHON LIBRARIES

- **Numpy** - This library is used to do Mathematical operations using arrays, matrices, etc...
- **Pandas** - It is used for data analysis and allows to import dataset.
- **Matplotlib** - It is used for data visualization and graphical plotting.
- **Seaborn (SnS)** - It is used to plot statistical graphics.
- **Nltk** - Natural Language Toolkit is used for text processing like tokenization, classification, tagging
- **Word cloud** – It is used to visualize the highlighted words in the text.
- **Text blob** - It is used to analyse the textual data and find part-of-speech tagging, sentiment, translation.
- **geopandas** – It is used to work with geospatial data in Python.
- **shapely.geometry** - Geometric objects can be used in Python with Shapely. Shapely is used for doing various geometric operations. The most fundamental geometric objects are Points, Lines and Polygons.
- **Unicode data** - This module provides access to the Unicode Character Database (UCD) which defines character properties for all Unicode characters.
- **String** – It contains number of functions to process string operations.
- **re** – It is used to identify the error while compiling.
- **io** – It allows to manage the file-related input and output operations.

CHAPTER-2

DATA COLLECTION AND DATA CLEANING

2.1 DATA

The collection of raw facts and figures is called data. The word raw means the true fact which is still not processed to get the exact meaning. It is collected from different sources used for different purposes such as number, characters, symbols, etc. These can also be included in data.

2.2 TYPES OF DATA

- Quantitative Data
- Qualitative Data

2.2.1 QUANTITATIVE DATA

Quantitative data is a kind of information which could be counted or measured. It is commonly used by researchers when they need to quantify a problem or else to rise any questions like what, how many. It is often used in math calculations, algorithms, etc. It is of two types,

- Discrete data
- Continuous data

Example: Distance, Number of weeks in years, Measurements, Costs and Age are some examples of the Quantitative data

2.2.2 QUALITATIVE DATA

Qualitative data is a kind of information which cannot be easily counted or measured or expressed using numbers or variables. It is collected from the text, image and audio through data visualization tools which includes graphs, maps, timelines etc. It remains unstructured. It can be collected through direct or indirect observation. It is of two types,

- Nominal data
- Ordinal data

For Example: If the student is reading a paragraph from a book during a period, the teacher who listens to the student finally gives the feedback on how the student had read that paragraph. If the teacher gives feedback based on fluency, and the clarity in pronunciation then it is considered as an example of qualitative data.

2.3 DATA COLLECTION

2.3.1 INTRODUCTION

Data constitutes the base and the findings of an investigation depends upon the correctness and completeness of the relevant data. The Sources of data are of two kinds namely primary source and secondary source. The term source means origin or the place from which data is extracted. According to George Simpson and Fritz Kafka, "A primary source is one that itself collects the data and a secondary source is one that makes available data which were collected by some other agency" [6]. Here data are collected from Twitter to perform the analysis. In particular, sports data are collected from the Twitter using the following ways.

2.3.2 VARIOUS WAY OF DATA COLLECTION

- Application Programming Interface (Twitter API)
- External websites (Kaggle)
- Manual data collection

Application Programming Interface (Twitter API)

Using this method, one could collect all the tweets from the Twitter. For this particular cause, an API application need to be created and then product keys like consumer key, consumer secret key, access token key and access token secret key need to be generated. By using these keys, one could be able to authenticate the Twitter website and extract the required tweets by using the hashtag or keyword in the suitable Python program.

External websites

There are many kinds of external websites to extract the datasets. Some of them include Kaggle, Google custom dataset search, UCI machine learning repository, and data.gov. The advantage of using these external websites is that there is no permission required in accessing the data. Here we have used Kaggle [10] to collect data as it is user friendly.

Manual data collection

Manual data collection is one of the toughest ways in collecting the data. It requires lot of time. Mostly this type of data extraction method is used when the amount of data to be collected is less. It is also possible to end up with some errors in this type of data extraction since it is done manually. But it can be customised as per our requirement. Data cleaning may take very less time compared to the above said cases.

2.4 DATA PROCESSING

The collection of raw data is the most important step of data processing. The collection of raw data type has major impact on the output produced. Data should be collected from defined sources in such a way that the findings are usable and valid.

- Define the aim of the research
- Choose the data collection method
- Plan the data collection procedures
- Collect the data
- Preparation

2.4.1 Define the aim of your research

We need to identify, what we have to achieve before collecting the data. We need to collect the data based on the research question whether it is qualitative or quantitative. We need to collect qualitative data, to test the hypothesis and collect qualitative data, to explore ideas and to understand the experience. If there are several aims, then we can use mixed method to collect the types of data.

2.4.2 Choose your data collection method

There are several methods involved for collecting the data, which includes

- Experimental research is a quantitative method.
- Interview focus groups and ethnographic is a qualitative method.
- Survey, observations, secondary data collection will serve as both qualitative or quantitative method.

2.4.3 Plan your data collection

We will come to know which method is used by keeping we will exactly.

2.4.4 Collect the data

Finally, we can implement our chosen methods to measure or observe the variables.

2.4.5 Preparation

Data preparation is the process of filtering the raw data in order to remove the unwanted and inaccurate data. It checks for the errors, duplications and miscalculations. The purpose of this step is to remove the data which is not good, so that by assembling high quality information and it can be used in best possible way.

2.5 DATA CLEANING

2.5.1 Introduction

Data cleaning is a process of detecting and correcting the unnecessary records from a recorded set or data. In data cleaning, incorrect and irrelevant parts of the data are identified and then, removed by using methods like replacing, modifying, etc... In simple, we say Data cleaning is the process of fixing, removing incorrect, corrupted or incomplete data from a dataset. Once the cleaning process is done, the data in the dataset are becomes fit to use in a program.

2.5.2 Necessity of data cleaning

Data cleaning is very much important to ensure that we have to achieve high data integrity. It removes errors, increase data reliability, ensure consistency, etc... If a dataset is processed before the cleaning process, then there will be a larger difference between the result obtained and the actual result. This process helps many people in many ways. For example, this process paves the way to get the recent information about the particular thing/paperwork searched by the individual by eliminating the past records. That is the most recent files and important document will be shown first.

2.5.3 Steps of data cleaning

Remove duplicate or irrelevant observations

Remove unwanted observation from your dataset, including duplicate observations irrelevant observations. During the data collection duplicate observation will often happen.

Ignoring the tuple

For a small database, ignoring the tuple may lead to the loss of important information for the analysis. For a large database we will be manually filling the missing values as it is a time-consuming process.

Fix structural errors

Structural errors occur when we measure or transfer data and notice strange naming conventions, types or incorrect capitalization. For Example: Although “N/A” and “Not Applicable” are different text, they should be analyzed as the same category.

Handle missing values

Missing data can’t be ignored because many algorithms will not accept missing values. There are a couple of ways to deal with missing data. Though both the ways alter the result to some extent it can be considered. One way is by dropping the observations that have missing values, but by doing this there is a chance of losing the information. So, one must be careful before removing it. The other way is by inputting the missing values based on the other observations. In this case also, there is an opportunity to lose the data because actual observations are not used instead assumptions are made. Another way is, to alter the way the data is used to effectively navigate null values.

Input

In this step, the raw data is converted into machine readable form and fed into the processing unit. This can be done in the form of data entry through a keyboard, scanner or any other input source.

Data processing

In this step, the raw data is subjected to various data processing methods using Machine Learning and Artificial Intelligence algorithms to generate a desirable output. This step may vary slightly from process to process depending on the source of data being processed (data lakes, online databases, connected devices, etc.) and the intended use of the output.

Output

The data is finally transmitted and displayed to the user in a readable form like graphs, tables, vector files, audio, video, documents, etc. This output can be stored and further processed in the next data.

2.6 EXAMPLE

The following programs helps in understanding the data cleaning or pre-processing process. A data set of a few tweets about the IPL containing emojis, hyperlink, URL etc. is taken and the pre-processing steps are carried out.

Step 1: Reading and printing the dataset

The following snippet is used to read and print the dataset in order to proceed with the pre-processing process.

```
1   import pandas as pd
2   df = pd.read_csv('IplTweet.csv')
3   print(df)
```

```
                                                tweet
0  MI vs DC Dream11 Team Prediction  Visit :  htt...
1  #istanbulbasaksehor vs #ManchesterUtd free dre...
2  #Velocity vs #supernovas  Free dream 11 team c...
3  Which team will reach the IPL 2020 finals firs...
4  @TamilarasanMut3 @StarSportsTamil They r askin...
5                              #Dream11IPL2020 😐🚀
```

Step 2: Removing Punctuations

Punctuations need to be removed in the pre-processing steps before starting the program, for this purpose the below snippet is used. It is simply used to treat all the words as the same.

```
import string
PUNCT_TO_REMOVE = string.punctuation
def remove_punctuation(tweet):
    return tweet.translate(str.maketrans('', '', PUNCT_TO_REMOVE))
df["tweet"] = df["tweet"].apply(lambda tweet: remove_punctuation(tweet))
print(df)
```

```
                                               tweet
0  MI vs DC Dream11 Team Prediction  Visit   http...
1  istanbulbasaksehor vs ManchesterUtd free dream...
2  Velocity vs supernovas  Free dream 11 team cli...
3  Which team will reach the IPL 2020 finals firs...
4  TamilarasanMut3 StarSportsTamil They r asking ...
5                             Dream11IPL2020 😐🧎
```

Step 3: Converted to lower case

To convert all the words in the sentence into a lower case is very important and helpful because all the words are treated in the same way and to avoid the duplicate words from the sentences which in turn helps in reducing the number of distinct words in the text.

```
df['tweet'] = df['tweet'].str.lower()
print(df)
```

```
                                               tweet
0  mi vs dc dream11 team prediction  visit   http...
1  istanbulbasaksehor vs manchesterutd free dream...
2  velocity vs supernovas  free dream 11 team cli...
3  which team will reach the ipl 2020 finals firs...
4  tamilarasanmut3 starsportstamil they r asking ...
5                             dream11ipl2020 😐🧎
```

Step 4: Remove duplicates

Duplicates are nothing but the repeated words in the same sentences. Once the text is converted into lower case, it is possible for us to get the count of all the duplicate words and remove them completely from the program, for which we use the following code.

```
df = df.drop_duplicates()
print(df)
```

```
                                                tweet
0  MI vs DC Dream11 Team Prediction  Visit   http...
1  istanbulbasaksehor vs ManchesterUtd free dream...
2  Velocity vs supernovas  Free dream 11 team cli...
3  Which team will reach the IPL 2020 finals firs...
4  TamilarasanMut3 StarSportsTamil They r asking ...
5                              Dream11IPL2020 😐🧜‍♀️
```

Step 5: Removing Hyperlink

Hyperlinks and URLs in the text are not that important for an analysis because no useful information is gained from them. Removing them from the text is necessary to proceed with the analysis part. In order to remove them, the following snippet is used.

```
df['tweet'] = df['tweet'].replace(r'http\S+', '', regex=True).replace(r'www\S+', '', regex=True)
print(df)
```

```
                                          tweet
0  mi vs dc dream11 team prediction  visit    dr...
1  istanbulbasaksehor vs manchesterutd free dream...
2  velocity vs supernovas  free dream 11 team cli...
3  which team will reach the ipl 2020 finals firs...
4  tamilarasanmut3 starsportstamil they r asking ...
5                          dream11ipl2020 😐🧜‍♀️
```

Step 6: Remove stop words

Stop words are the commonly used words in the English language. a, an, the, it, but, I, you, you're, myself, my etc., are some of the stop words. Since it doesn't provide any information while doing the analysis it is better to remove them. An Nltk Library containing the stop words are first downloaded and then the following coding is used to remove the stop words completely.

```
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
", ".join(stopwords.words('english'))
STOPWORDS = set(stopwords.words('english'))
def remove_stopwords(tweet):
    return " ".join([word for word in str(tweet).split() if word not in STOPWORDS])
df["tweet"] = df["tweet"].apply(lambda tweet: remove_stopwords(tweet))
print(df)
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
                                               tweet
0  mi vs dc dream11 team prediction visit dream11...
1  istanbulbasaksehor vs manchesterutd free dream...
2  velocity vs supernovas free dream 11 team clic...
3  team reach ipl 2020 finals first mi ⭐ vs dc 🦁 ...
4  tamilarasanmut3 starsportstamil r asking dream...
5                              dream11ipl2020 😐🎗️
```

Step 7: Removing frequently used words

Frequent words in the text slower down the process of running the program. So, by removing the frequent words, the time of execution of the program can reduced. Also, it improves the efficiency of the program. To perform this, initially the frequent words are collected and then it is removed using the following coding.

```
from collections import Counter
cnt = Counter()
for text in df["tweet"].values:
    for word in text.split():
        cnt[word] += 1
cnt.most_common(10)
FREQWORDS = set([w for (w, wc) in cnt.most_common(10)])
def remove_freqwords(tweet):
    return " ".join([word for word in str(tweet).split() if word not in FREQWORDS])
df["tweet"] = df["tweet"].apply(lambda tweet: remove_freqwords(tweet))
print(df)
```

```
                                               tweet
0  prediction visit dream11ipl dream11team dream1...
1  istanbulbasaksehor manchesterutd click link pr...
2  velocity supernovas click link prediction 👇👇 d...
3  reach ipl 2020 finals first ⭐ 🦁                 ...
4        tamilarasanmut3 starsportstamil r asking 😂
5                                          😐🎗️
```

Step 8: Removing rare words

Rare words in the text are not so important because it does not bring any change while analysing the sentiment of the text. Similar to frequent words, rare words are removed by collecting them initially and then removing it by using the following snippet.

```
n_rare_words = 10
RAREWORDS = set([w for (w, wc) in cnt.most_common()[:-n_rare_words-1:-1]])
def remove_rarewords(tweet):
    return " ".join([word for word in str(tweet).split() if word not in RAREWORDS])
df["tweet"] = df["tweet"].apply(lambda tweet: remove_rarewords(tweet))
print(df)
```

```
                                               tweet
0  prediction visit dream11ipl dream11team dream1...
1  istanbulbasaksehor manchesterutd click link pr...
2  velocity supernovas click link prediction 👇👇 d...
3  reach ipl 2020 finals first ⭐ 🦁                 ...
4
5
```

Step 9: Removing Emojis

Emoji is also used to express emotion in a symbolic way. Though it expresses the emotion in some kind of way, it is not necessary while compiling the program as we deal only with text. So, in order to remove the emojis, the following snippet is used.

```
df= df.astype(str).apply(lambda x: x.str.encode('ascii', 'ignore').str.decode('ascii'))
print(df)
```

```
                                             tweet
0  prediction visit dream11ipl dream11team dream1...
1  istanbulbasaksehor manchesterutd click link pr...
2  velocity supernovas click link prediction  dre...
3  reach ipl 2020 finals first     follow tilltos...
4
5
```

CHAPTER 3

ANALYSIS OF DATA USING MACHINE LEARNING

3.1 MACHINE LEARNING

Machine learning is nothing but it is a study of a computer algorithm. Machine learning is the process of predicting results (output) using the given input data. It is seen as a part of data science. In Machine learning input data are separated into Training Data and Testing Data from which it built a model using training dataset and predicts the output of the testing dataset. Machine learning is classified into two types (i) Supervised Machine learning, (ii) Unsupervised Machine learning (iii) Reinforcement Learning. There are different types of Machine Learning Algorithms that are used to predict results. Machine learning is used in many business domains like banking, finance, insurance and healthcare etc.

3.2 TYPES OF MACHINE LEARNING

3.2.1 SUPERVISED MACHINE LEARNING

It is a Machine learning model where the models are trained using labelled data. This type needs external supervision to learn. This is also classified into two namely classification and regression.

EXAMPLE

Consider the coins of different countries, here by using features given to model we would identify which country the coin belongs to.

3.2.2 UNSUPERVISED MACHINE LEARNING

It is a Machine Learning method in which the patterns are unlabelled data. The goal is to find the pattern and structure of input data. It is difficult to understand. Unsupervised Machine Learning is classified into two namely clustering and association.

EXAMPLE

In a classroom without the help of the teacher, student learn their subjects by themselves by having discussion among them.

3.2.3 REINFORCEMENT LEARNING

Reinforcement Learning is used where there is no idea about the label of a particular data. This learning model learns and updates itself. It is more complex to understand and apply.

EXAMPLE

Robotics is the best example of reinforcement because it learns things on its own.

The Input Data is separated by

(i) Training data.

(ii) Testing data.

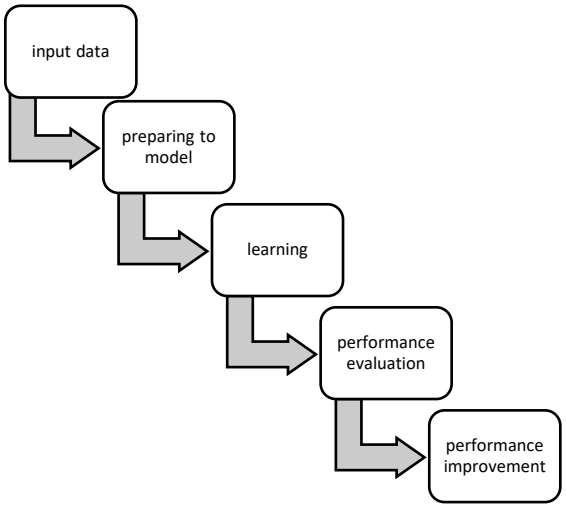
Training data

We know that in Machine learning dataset are split into two subset one is training dataset which use to train machine learning model.

Testing Data

Testing data is nothing but to test the pre trained data using programming language.

3.3 MACHINE LEARNING ACTIVITIES



3.4 TYPES OF MACHINE LEARNING ALGORITHMS

3.4.1 NAÏVE BAYES ALGORITHM

Naïve Bayes Algorithm is a Machine Learning Algorithm. It is the simple technique for building classifiers. Naïve means occurrence of certain feature is independent to the occurrence of other. Bayes is based on Bayes Theorem. The Bayes rule is that the outcome of hypothesis can be predicted by some evidence.

$$P(A/B) = P(B/A).\ P(A) \mid P(B)$$

P (A/B) – is the posterior probability.

P (B/A) – is the likelihood.

P (A) – is the prior probability.

Advantages of Naïve Bayes algorithm

- ❖ It is fast and easy to perform.
- ❖ It is suitable for binary and multi-class classification.
- ❖ It is mostly used in text classification.

Disadvantages of Naïve Bayes Algorithm

- ❖ It assumes that all features are independent.

Application of Naïve Bayes Classifier

- ❖ It is used for Credit scoring.
- ❖ It is used in medical data classification.
- ❖ It is used in text classification such as Spam filtering and Sentiment analysis.

3.4.2 SUPPORT VECTOR MACHINE (SVM)

Support Vector Machine creates decision boundary to separate data and in this SVM kernel and hyperplane are used.

Hyperplane

Hyperplane is a decision boundary, which separate data into two separate lines. When it has three features it would separate into two-dimensional plane.

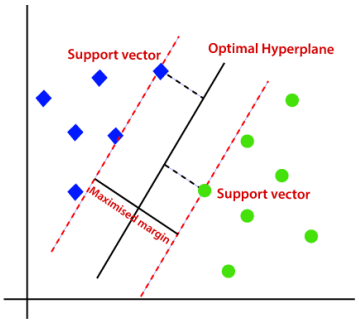
Kernel type

Kernel type is used to transform lower dimension data into higher dimension.

TYPES OF SVM

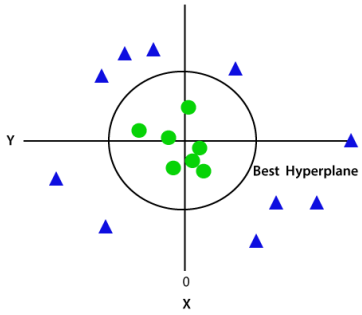
Linear SVM

Linear SVM is used for separately data where the dataset can be classified into two classes by using a single straight line.



Non-Linear SVM

In Non-Linear SVM is used for non-linearly separated data where the dataset cannot be classified using a straight line.



Advantages of SVM

- ❖ It works well when there is a clear margin of separation between classes.
- ❖ It is more effective in high dimensional spaces.
- ❖ It is relatively more efficient.

Disadvantages of SVM

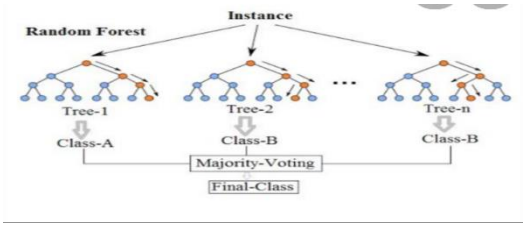
- ❖ It is not suitable for large data sets.
- ❖ It will not perform well when the data set has more noise.

Application of SVM

- ❖ Image-based analysis and classification tasks.
- ❖ Text-based applications.
- ❖ Computational biology.
- ❖ Security based application.

3.4.3 RANDOM FOREST MODEL

Random Forest model is nothing but a supervised Machine learning Algorithm, that is used widely in Classification and Regression problems. It build different decision trees on different samples.



Advantages of Random Forest

- ❖ It is capable of performing both classification and regression tasks.
- ❖ It is capable of handling high dimension data sets.

Disadvantages of Random Forest

- ❖ It is capable of performing both classification and regression tasks but it is not more suitable for regression tasks.

Application of Random Forest

- ❖ **Banking:** Banking sector mostly uses this algorithm for loan risk interpretation.

- **Land Use:** This algorithm is used to identify the similar areas.
- **Marketing:** Marketing trends can be identified using this algorithm.

3.5 NAÏVE BAYES ALGORITHMS

3.5.1 BAYES THEOREM [3]

If E1,E2, E3,…. En are mutually disjoint events with P(Ei) ≠ 0, i=1,2,3,…n then for any arbitrary events A which is a subset $\bigcup_{i=1}^{n} Ei$ such that P(A) > 0, we have

Formula

$P(Ei\backslash A) = \frac{P(Ei)P(A\backslash Ei)}{P(A)}$ where i=1,2,3,…..n

Ei & A = event.

P(Ei\A) = probability of Ei given A is true.

P (A/Ei) = probability of A given Ei is true.

P (Ei) & P (A) = the independent probabilities of Ei and A.

3.5.2 CONDITIONAL PROBABILITY

Conditional Probability is a measure of probability of event Ei occurred given that another event A has occurred.

Formula

P(Ei\A) = P(Ei∩A) / P(A) (or)

P (A/Ei) = P(Ei∩A) / P(Ei)

Example

The probability of selling a TV in a given normal day may be 30%. But if we consider that given day is Diwali, there will be probability of selling a TV will be more. The conditional probability of selling a TV on a day given that day might be Diwali is 70%.

3.5.3 INDEPENDENT EVENTS

The outcome of one event does not affect the outcome of other events. If Ei and A are the independent events then the probability of both occurring is

P (Ei∩A) =P (Ei).P (A)

Example

If we toss a coin in the air and get the outcome as head, then again if we toss a coin but time, we will get the outcome as tail. In both the cases, the occurrence of both events is independent of each other.

3.5.4 TYPES OF NAÏVE BAYES ALGORITHM

3.5.4(a) Gaussian Naïve Bayes Algorithm

It is used in classification and it assumes that feature follows normal distribution.

3.5.4(b) Multinomial Naïve Bayes Algorithm

It is used for discrete count and it is primarily used in the domains such as sports, politics, education etc.

3.5.4(c) Bernoulli Naïve Bayes Algorithm

The binomial model is used for feature where vectors are binary. This model is famous for document classification tasks.

Example

To check whether the player can play in all the Weather with respect to the data given:

	Outlook	Play
0	Rainy	Yes
1	Sunny	Yes
2	Overcast	Yes
3	Overcast	Yes
4	Sunny	No
5	Rainy	Yes
6	Sunny	Yes
7	Overcast	Yes

8	Rainy	No
9	Sunny	No
10	Sunny	Yes
11	Rainy	No
12	Overcast	Yes
13	Overcast	Yes

Given:

Yes- Sunny =3, No- Sunny =2

Yes- Rainy = 2, No- Rainy= 2

Yes- Overcast=5, No-Overcast=0

Frequency table for the Weather Conditions:

Weather	**Yes**	**No**
Overcast	5	0
Rainy	2	2
Sunny	3	2
Total	10	4

Likelihood table for Weather Conditions:

Weather	**No**	**Yes**	
Overcast	0	5	5/14=0.35
Rainy	2	2	4/14=0.29

Sunny	2	3	5/14=0.35
All	4/14=0.29	10/14=0.71	

Applying Bayes Theorem:

P (Yes/Sunny) = P (Sunny/Yes)*P (Yes)/P (Sunny)

P (Sunny/Yes) = 3/10=0.3

P (Sunny) = 0.35

P (Yes) = 0.71

So P (Yes/Sunny) = 0.3*0.71/0.35=0.60

P (No/Sunny) = P (Sunny/No)*P (No)/P (Sunny)

P (Sunny/No) = 2/4=0.5

P (No) = 0.29

P (Sunny) =0.35

So P (No/Sunny) = 0.5*0.29/0.35 = 0.41

So we can see from the above calculation that P (Yes/Sunny) > P (No/Sunny)

Hence on a Sunny day, Player can play the game.

P (Yes/Rainy) = P (Rainy/Yes)*P (Yes)/P (Sunny)

P (Rainy/Yes) = 2/10=0.2

P (Rainy) = 0.29

P (Yes) = 0.71

So, P (Yes/Sunny) = 0.2*0.71/0.29=0.48

P (No/Rainy) = P (Rainy/No) *P (No)/P (Rainy)

P (Rainy/No) = 2/10=0.2

P (No) =0.29

P (Rainy) =0.29

So, P (No/Rainy) = 0.2*0.29/0.29 = 0.2

P (Rainy) =0.29

So, we can see from the above calculation that P (Yes/Rainy) > P (No/Rainy)

Hence on a Rainy day, Player can play the game.

P (Yes/Overcast) = P (Overcast/Yes) *P (Yes)/P (Overcast)

P (Overcast/Yes) = 5/10 = 0.5

P (Overcast) = 0.35

P (Yes) = 0.71

So, P (Yes/Overcast) = 0.5*0.71/0.35

P (No/Overcast) = P (Overcast/No) *P (No)/P (Overcast)

P (Overcast/No) = 0

P (No) = 0.29

P (Overcast) =0.35

So, P (No/Overcast) =0*0.29/0.35 = 0

So, we can see from the above calculation that P (Yes/Overcast) > P (No/Overcast)

Hence on an Overcast day, Player can play the game.

CHAPTER 4

VARIOUS TYPES OF ANALYSIS

4.1 CLASSIFICATION OF DATA

Data classification is nothing but arranging the collected data into the class or subclass by their common characteristics. The well-planned data classification makes data easy to understand. This is a way to improve and maximize data security. It is used to reduce the volume of the data which minimizes the labor work.

4.2 TYPES OF CLASSIFICATION

There are four basic types used to classify the data:

1. Geographical classification
2. Qualitative classification
3. Quantitative classification
4. Chronological classification

4.2.1 GEOGRAPHICAL CLASSIFICATION

When the data is classified based on geographical locations such as countries, states, cities, etc., then such classification is called geographical classification. This is also known as spatial classification.

4.2.2 QUALITATIVE CLASSIFICATION

In this classification, data are the classified based on some attributes or qualities like honesty, intelligence, status, etc. It is interpretation-based and descriptive. It helps to understand the questions why, what, etc. Qualitative data are subjective and unique.

4.2.3 QUANTITATIVE CLASSIFICATION

This type of classification is made based on some measurable characteristics like height, length, weight, age, marks, etc. It is number-based and countable. It helps to understand how much, how many, etc. Quantitative data are universal and constant.

4.2.4 CHRONOLOGICAL CLASSIFICATION

It is possible to classify the data into group according to the timeline or alphabetical and this type of classification is known as chronological classification. In this case data are arranged in the order of year, quarters, months, weeks, etc. They can also be arranged according to alphabetical order. This type of classification is also known as temporal classification.

4.3 PROGRAM CODE

Now, we intend to analyse the data using the following python coding.

Step 1: Importing Python libraries

Python Libraries play an important role in including the special type of operations in the program. Since we intend to classify all the tweets based on the above types of data classification, certain libraries have to be included other than the basic ones.

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib.dates as mdates

# geodata
import geopandas as gpd
from shapely.geometry import Point

# custom y-axis
from matplotlib.ticker import FuncFormatter
def millions(x, pos):
    return '%1.1fM' % (x * 1e-6)

# ignoring warnings
import warnings
warnings.simplefilter("ignore")

# NLP
import unicodedata
import string
import re
import nltk
from nltk.corpus import stopwords
from nltk.util import ngrams
from wordcloud import WordCloud, STOPWORDS
from collections import Counter, defaultdict
from sklearn.feature_extraction.text import CountVectorizer
from textblob import TextBlob
from nltk.sentiment.vader import SentimentIntensityAnalyzer

# kaggle workspace
import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

Step 2: Importing and Reading dataset

To continue this program, the dataset has to be imported first and as mentioned earlier there are many ways in importing a dataset in a Google colab notebook. One such way is by importing it using the Google drive. And then, the data is read by using the following command.

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

```
Mounted at /content/drive
```

```
cricket = pd.read_csv('/content/drive/My Drive/T20_Worldcup_tweets.csv')
football = pd.read_csv('/content/drive/My Drive/FIFA.csv')
print('cricket dataset shape: {}'.format(cricket.shape))
print('football dataset shape: {}'.format(football.shape))
```

```
cricket dataset shape: (430383, 13)
football dataset shape: (530000, 16)
```

Step 3: Information in the dataset

Before proceeding with the program further, the information in the dataset need to be known. The function info () is used to get all the information like, number of columns listed along with their data types and number of data present.

```
cricket.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 430383 entries, 0 to 430382
Data columns (total 16 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   name         430014 non-null  object
 1   location     323089 non-null  object
 2   description  400116 non-null  object
 3   created      430381 non-null  object
 4   followers    430380 non-null  float64
 5   friends      430380 non-null  float64
 6   favourites   430380 non-null  float64
 7   verified     430380 non-null  object
 8   date         430380 non-null  datetime64[ns]
 9   text         430380 non-null  object
 10  hashtags     429971 non-null  object
 11  source       430379 non-null  object
 12  is_retweet   430379 non-null  object
 13  hour         430380 non-null  float64
 14  month        430380 non-null  float64
 15  day          430380 non-null  float64
dtypes: datetime64[ns](1), float64(6), object(9)
memory usage: 52.5+ MB
```

```
football.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 530000 entries, 0 to 529999
Data columns (total 19 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   id                530000 non-null  float64
 1   lang              530000 non-null  object
 2   Date              530000 non-null  datetime64[ns]
 3   Source            530000 non-null  object
 4   len               530000 non-null  int64
 5   Orig_Tweet        530000 non-null  object
 6   text              529449 non-null  object
 7   Likes             530000 non-null  int64
 8   RTs               530000 non-null  int64
 9   hashtags          468457 non-null  object
 10  UserMentionNames  455841 non-null  object
 11  UserMentionID     455841 non-null  object
 12  name              529945 non-null  object
 13  Location          390710 non-null  object
 14  followers         530000 non-null  int64
 15  friends           530000 non-null  int64
 16  hour1             530000 non-null  int64
 17  month1            530000 non-null  int64
 18  day1              530000 non-null  int64
dtypes: datetime64[ns](1), float64(1), int64(8), object(9)
memory usage: 76.8+ MB
```

Step 4: Printing the dataset

There are various ways to print the dataset, out of them we have chosen variablename.head(). When any specification is mentioned in the parenthesis then the rows according to the specification will be printed. For example, when head (2) is executed then only two corresponding rows will be printed.

Dataset of cricket

```
cricket.head()
```

	name	location	description	created	followers	friends	favourites	verified	date	text	hashtags	source
0	Prabhudatta Nayak????	Balangir, India	Proud to be an Indian ll #contestlover ll #bor...	21-05-2010 11:05	134.0	532.0	6625.0	0.0	2021-10-22 23:55:00	@ManappuramMAFIL Done Team \n@ManappuramMAFIL ...	['GuessAndWin', 'T20WorldCup', 'Contest', 'Cri...	Twitter for Android
1	Archisman Mishra	Bhubaneshwar, India	RISING FROM THE ASHES ENGINEER,GAMER,FOODY,POK...	10-12-2015 18:43	656.0	762.0	5286.0	0.0	2021-10-22 23:55:00	Set a reminder for my upcoming Space! https://...	['T20WorldCup', 'AUSvSA']	Twitter for Android
2	T20 World Cup	NaN	Official account of the ICC T20 World Cup. Men...	19-04-2018 12:46	378202.0	1097.0	296.0	1.0	2021-10-22 23:55:00	"We just try to enjoy everything that we do."\...	['WestIndies', 'T20WorldCup']	Khoros Publishing App
3	Farid Khan	Lahore, Pakistan	Journalist. Head of Digital Media @_cricingif ...	25-07-2021 03:59	1125.0	424.0	56.0	0.0	2021-10-22 23:54:00	#Pakistan and #India played each other in open...	['Pakistan', 'India', 'IND', 'T20WorldCup', 'NZ']	Twitter Web App
4	Bimal Mirwani	Hong Kong	I write all about Pakistan cricket on my site ...	01-03-2014 20:41	742.0	1433.0	811.0	0.0	2021-10-22 23:49:00	#Pakistan won't be much of a challenge for #In...	['Pakistan', 'India', 'Agarkar', 'AjitAgarkar'...	Twitter Web App

Dataset of Football

```
football.head()
```

	id	lang	Date	Source	len	Orig_Tweet	text	Likes	RTs	hashtags	UserMentionNames	UserMentionID	name
0	1.013600e+18	en	2018-02-07 01:35:00	Twitter for Android	140	RT @Squawka: Only two goalkeepers have saved t...	Only two goalkeepers have saved three penaltie...	0	477	WorldCup,POR,ENG	Squawka Football	Squawka	Cayleb
1	1.013600e+18	en	2018-02-07 01:35:00	Twitter for Android	139	RT @FCBarcelona: ?? @ivanrakitic scores the wi...	scores the winning penalty to send into the qu...	0	1031	WorldCup	FC Barcelona,Ivan Rakitic,HNS | CFF	FCBarcelona,ivanrakitic,HNS_CFF	Febri Aditya
2	1.013600e+18	en	2018-02-07 01:35:00	Twitter for Android	107	RT @javierfernandez: Tonight we have big game....	Tonight we have big game	0	488	worldcup	Javier Fernandez,Evgeni Plushenko	javierfernandez,EvgeniPlushenko	??
3	1.013600e+18	en	2018-02-07 01:35:00	Twitter Web Client	142	We get stronger\r\nTurn the music up now\r\nWe...	We get stronger Turn the music up now We got t...	0	0	PowerByEXO,WorldCup,FIFAStadiumDJ,XiuminLeague	EXO,FIFA World Cup ?	weareoneEXO,FIFAWorldCup	Frida Carrillo
4	1.013600e+18	en	2018-02-07 01:35:00	Twitter for Android	140	RT @Squawka: Only two goalkeepers have saved t...	Only two goalkeepers have saved three penaltie...	0	477	WorldCup,POR,ENG	Squawka Football	Squawka	tar

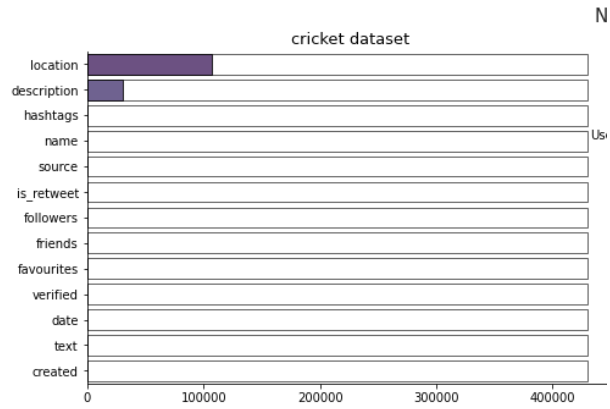
Step 5: Treating Missing Values

It is very important to treat the missing values before using some special operations in the text**.** **isna()** function is used to check whether there are any missing values present or not. If in case a missing value is found, then the particular row is deleted to proceed further. The number of missing values in the dataset can be witnessed from the bar diagram given below.

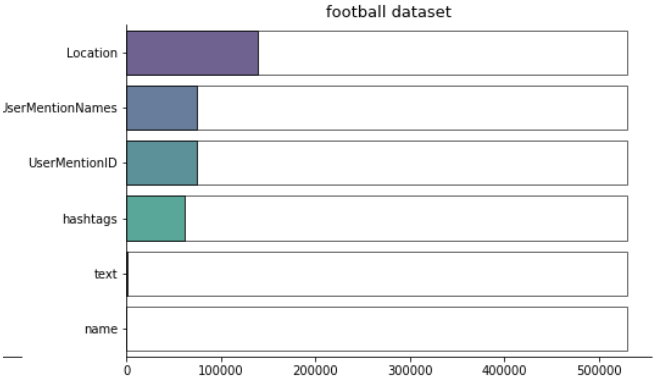
```
cricket_nan = pd.Series(cricket.isna().sum()[cricket.isna().sum() > 0].
                        sort_values(ascending = False))
football_nan = pd.Series(football.isna().sum()[football.isna().sum() > 0].
                        sort_values(ascending = False))

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(17, 5))
sns.set_style("whitegrid")
fig.suptitle('NaN values', size = 15)

sns.barplot(y = cricket_nan.index, x = [len(cricket)] * len(cricket_nan),
            edgecolor = 'black', color = 'white', alpha = 0.6, ax = ax1)
sns.barplot(y = cricket_nan.index, x = cricket_nan,
            edgecolor = 'black', alpha = 0.8, ax = ax1,
            palette = sns.color_palette("viridis", len(cricket_nan)))
ax1.get_xaxis().get_major_formatter().set_scientific(False)
ax1.set_title('cricket dataset', size = 13)
```

```
sns.barplot(y = football_nan.index, x = [len(football)] * len(football_nan),
            edgecolor = 'black', color = 'white', alpha = 0.6, ax = ax2)
sns.barplot(y = football_nan.index, x = football_nan,
            edgecolor = 'black', alpha = 0.8, ax = ax2,
            palette = sns.color_palette("viridis", len(football_nan)))
ax2.get_xaxis().get_major_formatter().set_scientific(False)
ax2.set_title('football dataset', size = 13)

sns.despine()
```

Output



NaN values

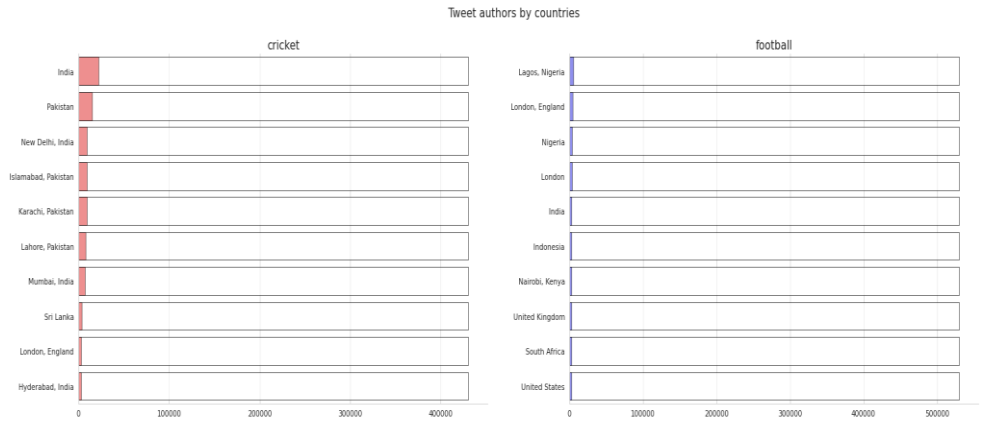


Step 6 Geographical Classification

The below snippet is used to find which sport is the most popular one based on the location. For this purpose, initially location column is extracted and then, based on the number of tweets by people in different countries, a bar diagram is drawn to find out the popularity.

```
cricket_tweets_location = cricket.location.value_counts()[:10]
football_tweets_Location = football.Location.value_counts()[:10]

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(25, 8))
sns.set_style("whitegrid")
fig.suptitle("Tweet authors by countries", size = 15)

sns.barplot(y = cricket_tweets_location.index,
            x = [len(cricket)] * len(cricket_tweets_location),
            edgecolor = 'black', color = 'white', alpha = 0.6, ax = ax1)
sns.barplot(y = cricket_tweets_location.index,
            x = cricket_tweets_location,
            edgecolor = 'black', color = 'red', alpha = 0.5, ax = ax1)
ax1.get_xaxis().get_major_formatter().set_scientific(False)
ax1.set_xlabel('')
ax1.set_title('cricket', size = 15)


sns.barplot(y = football_tweets_Location.index,
            x = [len(football)] * len(football_tweets_Location),
           edgecolor = 'black', color = 'white', alpha = 0.6, ax = ax2)
sns.barplot(y = football_tweets_Location.index,
            x = football_tweets_Location,
            edgecolor = 'black', color = 'blue', alpha = 0.5, ax = ax2)
ax2.get_xaxis().get_major_formatter().set_scientific(False)
ax2.set_xlabel('')
ax2.set_title('football', size = 15)

sns.despine()
```

Output



Step 7 Qualitative classification:

The following program code is used to collect all the positively highlighted words in the tweets and visualize them by using the word cloud. This type of classification comes under the qualitative one, since these highlighted words expresses the quality of the tweets.

```
show_wordcloud(df.loc[df['text_sentiment']=='Positive', 'text'], title = 'Prevalent words in texts (Positive sentiment)')
```

Cricket dataset

Prevalent words in texts (Positive sentiment)

Football dataset

Prevalent words in texts (Positive sentiment)

Step 8 Quantitative Classification

The quantitative classification is done in two ways. One is by considering the length of the tweets. That is, the length of each tweet in both the dataset is taken into account and finally the tweet which has the largest length is found.

```
cricket['text'][19529]
```

```
'?????????? Australia vs Sri Lanka Predictions &amp;amp; Tips - Smith tipped to be the difference for Australia\n\n?? Get our preview and betting tips! ??\n\n?? http
s://t.co/tVKRaLPsgn\n\n#T20WorldCup #tips #freetips ?? 18+ BeGambleAware'
```

```
football['text'][11650]
```

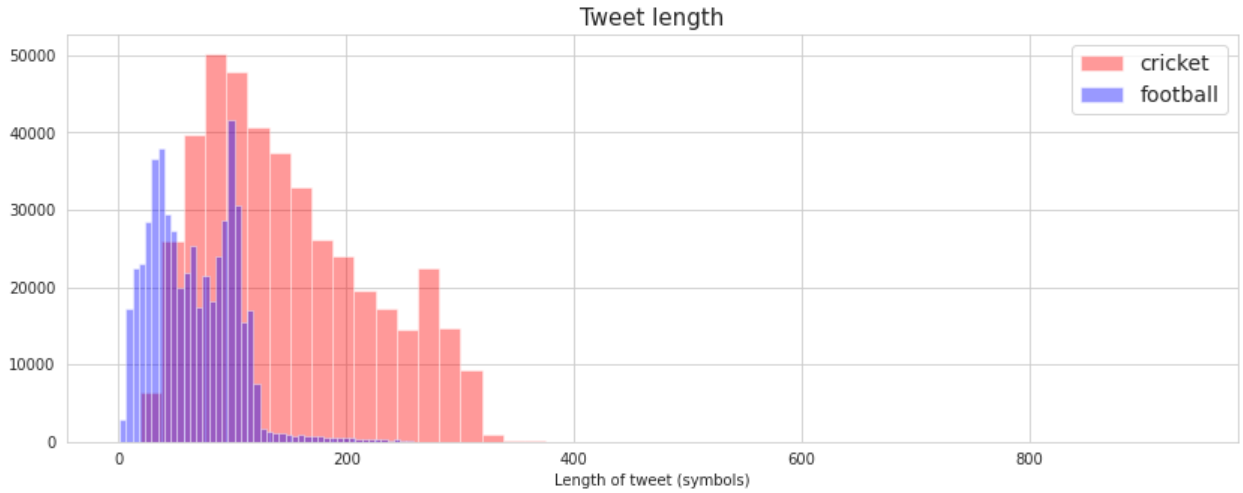
```
'It looks like Miguel Layun and Chicharito will be slightly more blonde when Mexico take on Brazil tomorrow'
```

```
# Tweet cleaner
def tweet_cleaner(text):
    return text
```

```
cricket_cleaned = cricket.copy()
football_cleaned = football.copy()

cricket_cleaned['text'] = cricket_cleaned['text'].apply(lambda x:tweet_cleaner(x))
football_cleaned['text'] = football_cleaned['text'].apply(lambda x:tweet_cleaner(x))
```

```
cricket_text_length = cricket_cleaned.text.str.len()
football_text_length = football_cleaned.text.str.len()

sns.set_style("whitegrid")
plt.figure(figsize=(14, 5))

sns.distplot(cricket_text_length, label = 'cricket', color = 'red', kde = False)
sns.distplot(football_text_length, label = 'football', color = 'blue', kde = False)
plt.legend(prop={'size': 14})
plt.title('Tweet length', size = 15)
plt.xlabel('Length of tweet (symbols)')
plt.show()
```

Output

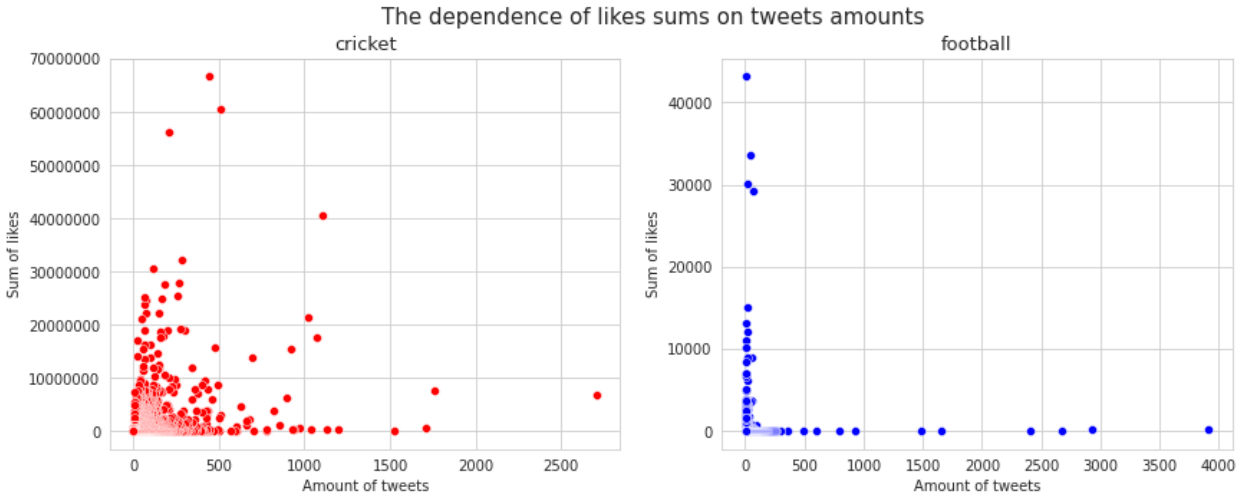


The other one is done by taking the number of likes into account. The likes column for each of the tweet is taken and among them the most liked tweet is found which brings us to the end of this quantitative classification. Also, a visualization is done which shows us the number of tweets and the major likes of the tweet.

```
cricket_count = pd.DataFrame(cricket['name'].value_counts())
cricket_count = pd.DataFrame({'name': cricket_count.index,
                              'count': cricket_count.name})
cricket_likes = cricket[['name', 'favourites']].groupby('name').sum()
cricket_agg = pd.merge(cricket_count, cricket_likes, on = 'name',
                       how = 'left')


football_count = pd.DataFrame(football['name'].value_counts())
football_count = pd.DataFrame({'name': football_count.index,
                              'count': football_count.name})
football_likes = football[['name', 'Likes']].groupby('name').sum()
football_agg = pd.merge(football_count, football_likes, on = 'name',
                       how = 'left')
```

```
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(14, 5))
sns.set_style("whitegrid")
fig.suptitle("The dependence of likes sums on tweets amounts", size = 15)

sns.scatterplot(x = cricket_agg['count'], y = cricket_agg['favourites'],
                color = 'red', ax = ax1)
ax1.get_yaxis().get_major_formatter().set_scientific(False)
ax1.set_xlabel('Amount of tweets')
ax1.set_ylabel('Sum of likes')
ax1.set_title('cricket', size = 13)


sns.scatterplot(x = football_agg['count'], y = football_agg['Likes'],
                color = 'blue', ax = ax2)
ax2.get_yaxis().get_major_formatter().set_scientific(False)
ax2.set_xlabel('Amount of tweets')
ax2.set_ylabel('Sum of likes')
ax2.set_title('football', size = 13)

fig.show()
```

Output



Step 9 Chronological Classification

The date column in both the data set is first taken and a further classification of that into month, day and hour is done. Then the tweets tweeted in the hourly basis is taken into account and is visualized by using a bar diagram.

```
cricket['date'] = pd.to_datetime(cricket['date'])
cricket['hour'] = cricket['date'].apply(lambda x: x.hour)
cricket['month'] = cricket['date'].apply(lambda x: x.month)
cricket['day'] = cricket['date'].apply(lambda x: x.day)
cricket.head()
```

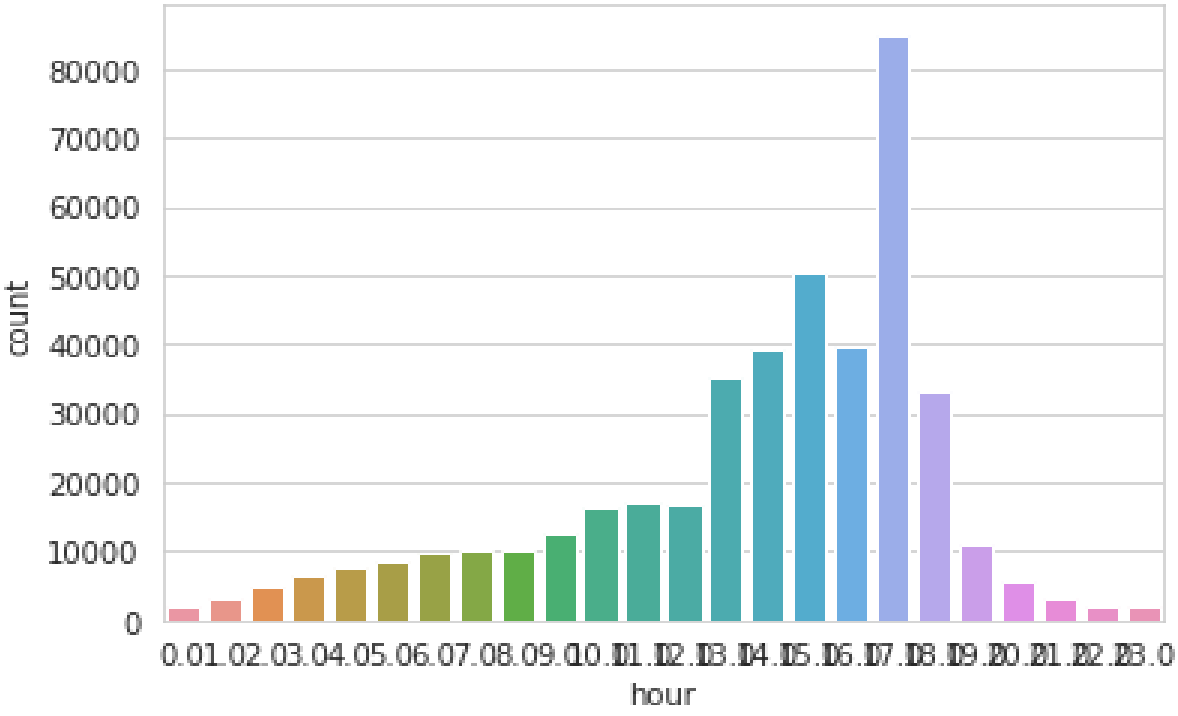
```
football['Date'] = pd.to_datetime(football['Date'])
football['hour1'] = football['Date'].apply(lambda x: x.hour)
football['month1'] = football['Date'].apply(lambda x: x.month)
football['day1'] = football['Date'].apply(lambda x: x.day)
football.head()
```

```
sns.countplot(x='hour', data = cricket)
```

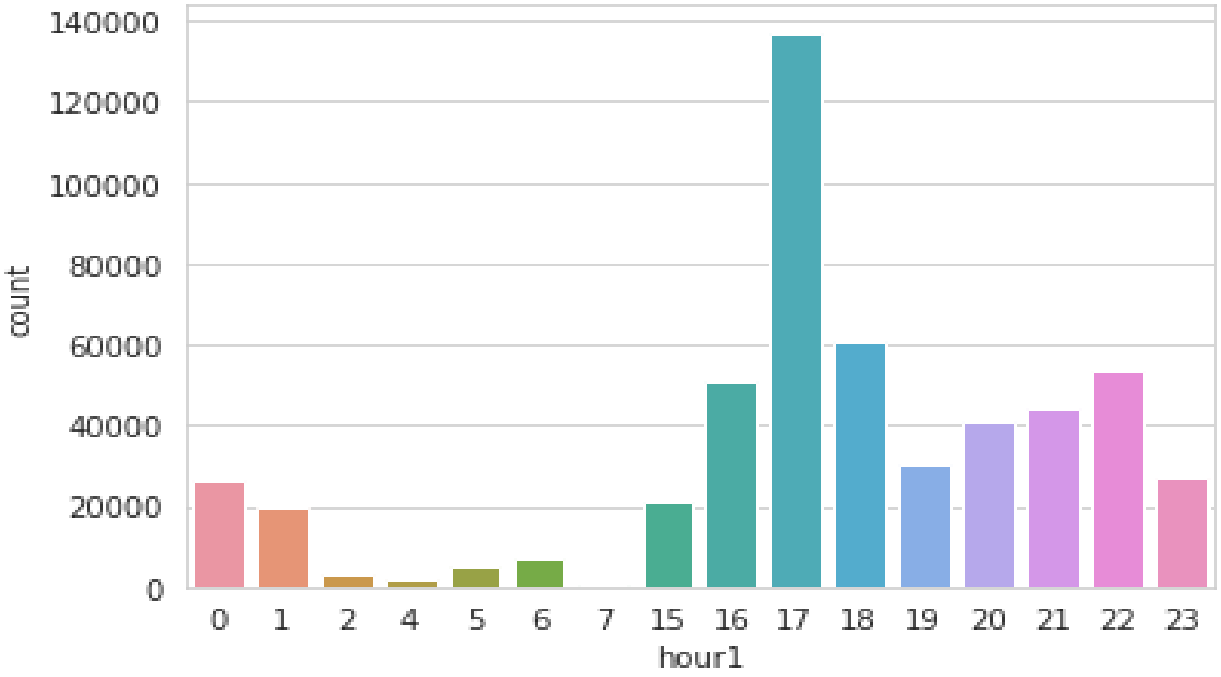
```
sns.countplot(x='hour1', data = football)
```

Output

Cricket dataset



Football dataset



4.4 INTERPRETATION FROM THE ABOVE ANALYSIS

By classifying the data based on geographical, qualitative, quantitative and chronological, we have arrived at the following results:

1. Using Geographical classification, we have checked for the common countries in both the dataset and we could find that in both the dataset India and England are in common. The tweets about cricket are from India and the tweets about Football are more from England. Thus, it is obvious that many people in India like to play and watch Cricket and Football and by comparing the total number of tweets based on both the locations, we could find that Cricket takes the lead.
2. By using Qualitative classification, we have taken the words that express the quality in both the dataset and visualized them. We could find the words like “Well played”, “Well done”, “Congratulations”, and “Won” which express the positivity.
3. We have done the Quantitative classification in two ways. One is by considering the length of the tweet and the other is by considering the number of likes that the tweets had got. Tweets about Cricket had got more number of likes than the tweets about Football. When

we considered the length of the tweet, we could find that the tweets about Cricket were expressed with more number of words than that of Football.

4. Using Chronological classification, we have classified our data set into hourly basis. From the bar diagram portraying the hourly basis of both Cricket and Football, we could see that more number of tweets about both data sets comes after 3:00 p.m. This shows that people are watching the sports or spending time for sports mostly in the evening. On Further observation, we came to a conclusion that Cricket is seen by the people or people talk about Cricket all the time whereas that is not the scenario for Football. Further, since the tweets in both the dataset are mostly from India, it is evident that Indians spend more time for Cricket than Football.

CHAPTER 5

INTERPRETATION AND GRAPHICAL REPRESENTATION

5.1 INTRODUCTION

Sentiment analysis can be done in two ways, one is by using the Semantic way of approach and the other is by using the Machine learning way. The Semantic approach uses the dictionary of words to find out the Sentiment of the given text. It is advisable to use the Semantic approach for a minimum amount of data. When the dataset is large, Machine learning approach is used in order to reduce the complexity of the task. Depending upon the dataset given, the type of Machine Learning algorithm (section 3.4) is chosen. Mostly the Supervised Machine Learning algorithm will be used for this type of analysis, since a labelled data will be chosen

5.2 INTRODUCTION TO DOMAIN

The domain chosen to analyse the Sentiment is Sports. One of the ways to relax or to entertain oneself is by playing or by watching games. It just boosts up one's energy to the maximum and refreshes one's mind and soul. Other than the game, there are lots of emotions that can also be witnessed in the game. Sometime the people watching and the people playing could feel the same. Since it is a wider field, analysing the sentiment requires a large amount of data. Here the main focus is to analyse the Sentiment of the tweets in the sports domain and finding out the majority of the sport which is popular among the people and also to find out the ratio of sports liked by the people. For this purpose, considering the fact that sports is a wide field, the recent world cup dataset of Cricket and Football is taken into account and the analysis is carried out and the following results are interpreted. The required dataset is taken form the website, https://www.kaggle.com/.

5.3 DATASET EXPLANATION:

There are many sport played by the players all over the world. Among those sports, we intend to find the top sport across the globe. It is seen that Cricket, Football, Basket Ball, Volley Ball take the top places. Considering the availability of dataset, Cricket and Football is taken into account. In both the games, many different types of competitions and leagues are played by the players all over the world. Among those tournaments, the twitter dataset of the T20 World cup in Cricket which is held in

the year 2019 and the FIFA Club world cup held in the year 2018 is taken from the website for this analysis purpose.

Through this program, we are going to find the popular sport liked by the people. For this purpose, twitter dataset are taken from the website named, Kaggle, which is a place to explore large number of datasets that acts as a key tool in the field of Data Science. Since the domain we have chosen is sport, in particular, Cricket and Football is taken, their corresponding twitter dataset is downloaded by searching them over the website. After getting the csv file of the required dataset, the further process is carried out.

5.4 PROGRAM CODE:

Step 1: Importing Python libraries

Python libraries contain a list of functions. It is used in a program to avoid writing the scrap part in the program. It can be included in the program in many ways, one such way is by using the pip command. Here the required Libraries like, pandas, matplot, io, unicodedata, numpy, string, nltk are imported.

```
# Setup
!pip install -q wordcloud
import wordcloud

import nltk
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')

import pandas as pd
import matplotlib.pyplot as plt
import io
import unicodedata
import numpy as np
import re
import string
```

Step 2: Reading and printing the datasets

There are three commonly used ways in importing the dataset to a Google colab workspace. One such way is importing the dataset using the Google drive. To achieve this, the following command is used. Once the authentication work is done in the Google drive, the file will be ready to access.

Here initially both the datasets are uploaded in the Google drive and then the first two lines of the following command is executed. And then for each dataset separately the last two lines are executed. For the second dataset (i.e. Football), the third line command is replaced by "data_df = pd.read_csv('/content/drive/My Drive/FIFA.csv')".

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

```
Mounted at /content/drive
```

```
data_df = pd.read_csv('/content/drive/My Drive/T20_Worldcup_tweets.csv')
```

```
data_df.head()
```

Cricket dataset

	name	location	description	created	followers	friends	favourites	verified	date	text	hashtags	source	i
0	Prabhudatta Nayak????	Balangir, India	Proud to be an Indian ll #contestlover ll #bor...	21-05-2010 11:05	134.0	532.0	6625.0	0.0	22-10-2021 23:55	@ManappuramMAFIL Done Team \n@ManappuramMAFIL ...	['GuessAndWin', 'T20WorldCup', 'Contest', 'Cri...	Twitter for Android	
1	Archisman Mishra	Bhubaneshwar, India	RISING FROM THE ASHES ENGINEER,GAMER,FOODY,POK...	10-12-2015 18:43	656.0	762.0	5286.0	0.0	22-10-2021 23:55	Set a reminder for my upcoming Space! https://...	['T20WorldCup', 'AUSvSA']	Twitter for Android	
2	T20 World Cup	NaN	Official account of the ICC T20 World Cup. Men...	19-04-2018 12:46	378202.0	1097.0	296.0	1.0	22-10-2021 23:55	"We just try to enjoy everything that we do."\...	['WestIndies', 'T20WorldCup']	Khoros Publishing App	
3	Farid Khan	Lahore, Pakistan	Journalist. Head of Digital Media @_cricingif ...	25-07-2021 03:59	1125.0	424.0	56.0	0.0	22-10-2021 23:54	#Pakistan and #India played each other in open...	['Pakistan', 'India', 'IND', 'T20WorldCup', 'NZ']	Twitter Web App	
4	Bimal Mirwani	Hong Kong	I write all about Pakistan cricket on my site ...	01-03-2014 20:41	742.0	1433.0	811.0	0.0	22-10-2021 23:49	#Pakistan won't be much of a challenge for #In...	['Pakistan', 'India', 'Agarkar', 'AjitAgarkar'...	Twitter Web App	

Football dataset

	ID	lang	Date	Source	len	Orig_Tweet	Tweet	Likes	RTs	Hashtags	UserMentionNames	UserMentionID
0	1013597060640145408	en	2018-07-02 01:35:45	Twitter for Android	140	RT @Squawka: Only two goalkeepers have saved t...	Only two goalkeepers have saved three penaltie...	0.0	477.0	WorldCup,POR,ENG	Squawka Football	Squawka
1	1013597056219295744	en	2018-07-02 01:35:44	Twitter for Android	139	RT @FCBarcelona: ?? @ivanrakitic scores the wi...	scores the winning penalty to send into the qu...	0.0	1031.0	WorldCup	FC Barcelona,Ivan Rakitic,HNS | CFF	FCBarcelona,ivanrakitic,HNS_CFF
2	1013597047482544130	en	2018-07-02 01:35:42	Twitter for Android	107	RT @javierfernandez: Tonight we have big game....	Tonight we have big game	0.0	488.0	worldcup	Javier Fernandez,Evgeni Plushenko	javierfernandez,EvgeniPlushenko
3	1013597044198391808	en	2018-07-02 01:35:41	Twitter Web Client	142	We get stronger\r\nTurn the music up now\r\nWe...	We get stronger Turn the music up now We got t...	0.0	0.0	PowerByEXO,WorldCup,FIFAStadiumDJ,XiuminLeague	EXO,FIFA World Cup ?	weareoneEXO,FIFAWorldCup
4	1013597039999926272	en	2018-07-02 01:35:40	Twitter for Android	140	RT @Squawka: Only two goalkeepers have saved t...	Only two goalkeepers have saved three penaltie...	0.0	477.0	WorldCup,POR,ENG	Squawka Football	Squawka

Step 3: Dataset information

This line of code is executed in order to find out the number of columns present in each dataset and also the total number of rows in each columns along with their data type.

```
data_df.info()
```

Cricket dataset

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 430383 entries, 0 to 430382
Data columns (total 13 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   name         430014 non-null  object
 1   location     323089 non-null  object
 2   description  400116 non-null  object
 3   created      430381 non-null  object
 4   followers    430380 non-null  float64
 5   friends      430380 non-null  float64
 6   favourites   430380 non-null  float64
 7   verified     430380 non-null  object
 8   date         430380 non-null  object
 9   text         430380 non-null  object
 10  hashtags     429971 non-null  object
 11  source       430379 non-null  object
 12  is_retweet   430379 non-null  object
dtypes: float64(3), object(10)
memory usage: 42.7+ MB
```

Football dataset

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 148194 entries, 0 to 148193
Data columns (total 16 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   ID                148194 non-null  int64
 1   lang              148194 non-null  object
 2   Date              148194 non-null  object
 3   Source            148194 non-null  object
 4   len               148194 non-null  int64
 5   Orig_Tweet        148194 non-null  object
 6   Tweet             148146 non-null  object
 7   Likes             148193 non-null  float64
 8   RTs               148193 non-null  float64
 9   Hashtags          127566 non-null  object
 10  UserMentionNames  125451 non-null  object
 11  UserMentionID     125451 non-null  object
 12  Name              148176 non-null  object
 13  Place             106200 non-null  object
 14  Followers         148193 non-null  float64
 15  Friends           148193 non-null  float64
dtypes: float64(4), int64(2), object(10)
memory usage: 18.1+ MB
```

Step 4: Finding Missing data

Missing data are common in all the datasets. The missing data in the dataset should be handled properly to reduce the error in the program. If any missing data is found in any column, the corresponding row should be removed so that a proper and appropriate result will be found. The following program code is used to find out the number of missing data in each column and the result for both the dataset is shown below.

```
def missing_data(data):
    total = data.isnull().sum()
    percent = (data.isnull().sum()/data.isnull().count()*100)
    tt = pd.concat([total, percent], axis=1, keys=['Total', 'Percent'])
    types = []
    for col in data.columns:
        dtype = str(data[col].dtype)
        types.append(dtype)
    tt['Types'] = types
    return(np.transpose(tt))
```

```
missing_data(data_df)
```

Cricket dataset

	name	location	description	created	followers	friends	favourites	verified	date	text	hashtags	source	is_retweet
Total	369	107294	30267	2	3	3	3	3	3	3	412	4	4
Percent	0.085738	24.929888	7.032573	0.000465	0.000697	0.000697	0.000697	0.000697	0.000697	0.000697	0.095729	0.000929	0.000929
Types	object	object	object	object	float64	float64	float64	object	object	object	object	object	object

Football dataset

	ID	lang	Date	Source	len	Orig_Tweet	Tweet	Likes	RTs	Hashtags	UserMentionNames	UserMentionID	Name	Place	Followers	Friends
Total	0	0	0	0	0	0	48	1	1	20628	22743	22743	18	41994	1	1
Percent	0.0	0.0	0.0	0.0	0.0	0.0	0.03239	0.000675	0.000675	13.919592	15.346775	15.346775	0.012146	28.33718	0.000675	0.000675
Types	int64	object	object	object	int64	object	object	float64	float64	object	object	object	object	object	float64	float64

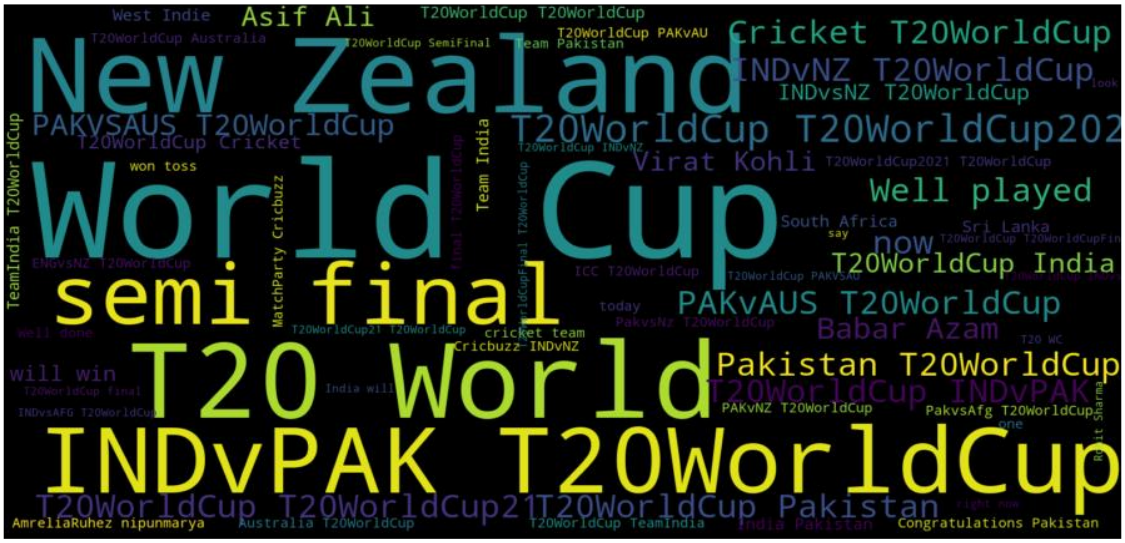
Step 5: Forming Word cloud

To visualize the words in the text, many types of Visualization techniques are available. One such is, Word cloud. Here the most frequent words are taken from the datasets and the word cloud visualization is done by using the following coding. This helps in knowing the highlighted words in the text.

```
def show_wordcloud(data, title=""):
    text = " ".join(t for t in data.dropna())
    stopwords = set(STOPWORDS)
    wordcloud = WordCloud(stopwords=stopwords, scale=4, max_font_size=50, max_words=500,background_color="black").generate(text)
    fig = plt.figure(1, figsize=(16,16))
    plt.axis('off')
    fig.suptitle(title, fontsize=20)
    fig.subplots_adjust(top=2.3)
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.show()
```

```
show_wordcloud(data_df['text'], title = 'Prevalent words in text')
```

Cricket



Prevalent words in text

Football

Prevalent words in text

Step 6: Text Pre-processing

After the visualization process, the tweets in the datasets are pre-processed as mentioned in (section 2.6)

Step 7: Finding the Sentiment

The final and the main step in this process is finding out the sentiment analysis. This is done by using the sentiment intensity analyser. Further the texts are classified into Positive, Negative and Neutral and is visualized with the help of the bar diagram.

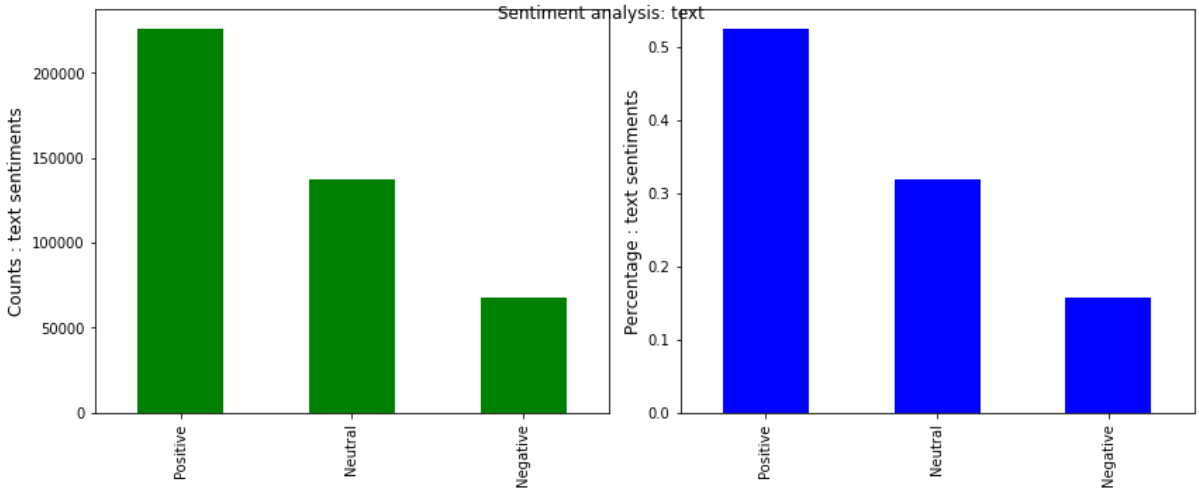
```
import nltk
nltk.download('vader_lexicon')
sia = SentimentIntensityAnalyzer()
def find_sentiment(post):
    if sia.polarity_scores(post)["compound"] > 0:
        return "Positive"
    elif sia.polarity_scores(post)["compound"] < 0:
        return "Negative"
    else:
        return "Neutral"
```

```
[nltk_data] Downloading package vader_lexicon to /root/nltk_data...
```

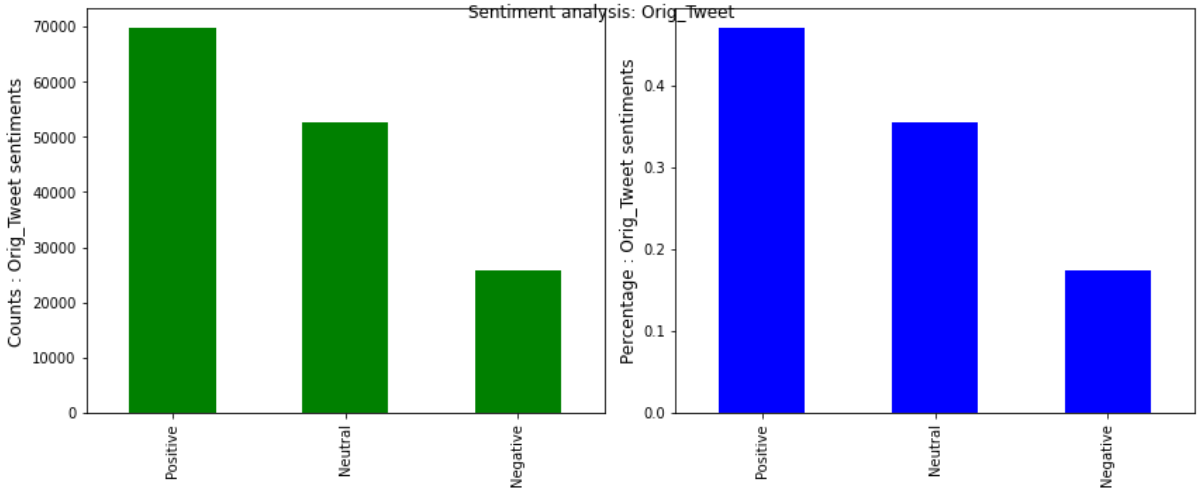
```
def plot_sentiment(df, feature, title):
    counts = df[feature].value_counts()
    percent = counts/sum(counts)

    fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(12, 5))

    counts.plot(kind='bar', ax=ax1, color='green')
    percent.plot(kind='bar', ax=ax2, color='blue')
    ax1.set_ylabel(f'Counts : {title} sentiments', size=12)
    ax2.set_ylabel(f'Percentage : {title} sentiments', size=12)
    plt.suptitle(f"Sentiment analysis: {title}")
    plt.tight_layout()
    plt.show()
```

```
def main():
 data_df['text_sentiment'] = data_df['text'].apply(lambda x: find_sentiment(str(x)))
 plot_sentiment(data_df, 'text_sentiment', 'text')
if __name__ == "__main__":
   try:
      main()
   except KeyboardInterrupt:
      # do nothing here
      pass
```

Cricket dataset



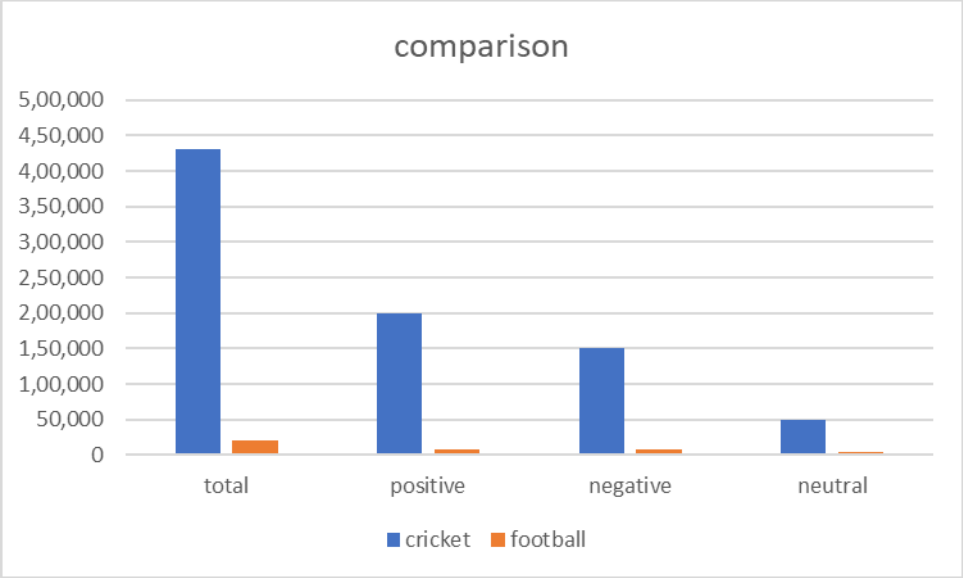
Football dataset



5.5 Interpretation

From the analysis done above, the following data is obtained and the comparison is made between them.

Polarity	Cricket (Approximate number of tweets)	Football (Approximate number of tweets)
Positive	2,10,800	8,500
Negative	1,60,800	8,600
Neutral	50,000	4,015
Total	4,30,380	21,115



In order to know about sport which is liked most by the people, the above comparison is made and then the positive tweets are considered. It is seen that Cricket is the mostly spoken sport and also liked by many people. In this case, Football stands next to Cricket.

CONCLUSION

We have studied the process of sentimental analysis in detail and analysed the sentiment of the Twitter datasets in sports domain. Sports is a wider field. Many kinds of analysis can be done in this field using the data which is available in many platforms. We have taken the recent world cup datasets of cricket and football which held in the year 2019 and 2018 respectively. Then we presented the data using a bar diagram, graph etc. On observing the data, we had come to the conclusion that Cricket is most popular than Football across the globe. Though this kind of analysis has some drawbacks, researchers and developers are finding ways to rectify and find solutions to the problem that people face during the process of Sentiment Analysis so that we can arrive at much more efficient results. Still deducting the emotions and analysing the sentiment is vital as that could enhance the digital marketing and advertisement. In the era of science and Technology, Machine Learning and AI are turning out to be boon.

REFERENCES

[1] Basant Agarwal, Namita Mittal (auth.) -Prominent Feature Extraction for Sentiment Analysis - Springer International Publishing (2016)

[2] Bing Liu - Sentiment Analysis_ Mining Opinions, Sentiments, and Emotions-Cambridge University Press (2015)

[3] Gupta S.P - Statistical Methods, sultan chand and sons publication, 2012 edition.

[4] Gupta.S.P and Gupta.M.P – Business Statistics

[5] Mark Summerfield - "Programming in Python 3", A Complete Introduction to the Python Language , pearson publication, second Edition.

[6] Navnitham P A - Business Mathematics and Statistics, jai publishers,

[7] Saikat dutt, Subramanian Chandramouli, Amit Kumar Das - "Machine Learning" Pearson Publications.

[8] Bo Pang and Lillian Lee Department - "Thumbs up? Sentiment Classification using Machine Learning Techniques", Proceeding of EMNLP, 2002.

[9] Vishal A. Kharde, S.S. Sonawane – "Sentiment analysis of Twitter Data: A

Survey of Techniques" International Journal of Computer Publications

Volume (139) No.11, April 2016.

[10] www.kaggle.com – for secondary data