**Project Report**

On

**DSBDAL Mini Project**

By

Giramkar Shrushti Ankush

Roll No: 19

Under the guidance

of

**Proff. Bhosale S.S**

Year of Submission: 2023-2024



HSBPVT’S

PARIKRAMA FACULTY OF ENGINEERING, KASHTI

DEPARMENT OF COMPUTER ENGINEERING

**HSBPVT’s GOI Faculty of Engineering, Kashti**

DEPARTMENT OF COMPUTER ENGINEERING

KASHTI



This is to certify that **Miss. Giramkar Shrushti Ankush** from **Third Year Computer Engineering** has successfully completed her seminar work titled **“DSBDAL Mini Project”** at HSBPVT’S of Engineering, Kashti in the partialfulfilment of the Bachelor’s Degree in Engineering of Savitribai Phule Pune University.

Proff. Bhosale.S. S Proff. Hiranwale.S.B

**Guide Name HOD Principal**

**INDEX**

|  |  |
| --- | --- |
| **Sr.NO** | **TITLE** |
| 1 | Problem Statement |
| 2 | Objective (Technology Used) |
| 3 | Introduction |
| 4 | Twitter Sentiment Analysis |
| 5 | Sentiment Classification |
| 6 | Block Diagram |
| 7 | Result Analysis |
| 8 | Case Study |
| 9 | Conclusion |

**PROBLEM STATEMENT:**

SENTIMENT ANALYSIS ON TWITTER

USING APACHE SPARK

SENTIMENT ANALYSIS ON TWITTER

USING APACHE SPARK

SENTIMENT ANALYSIS ON TWITTER

USING APACHE SPARK

SENTIMENT ANALYSIS ON TWITTER

USING APACHE SPARK

SENTIMENT ANALYSIS ON TWITTER

USING APACHE SPARK

SENTIMENT ANALYSIS ON TWITTER

USING APACHE SPARK

SENTIMENT ANALYSIS ON TWITTER

USING APACHE SPARK

Sentiment Analysis on Twitter using Apache Spark. In this project, we try to implement an NLP **Twitter sentiment analysis model** that helps to overcome the challenges of sentiment classification of tweets. We will be classifying the tweets into positive or negative sentiments.

**OBJECTIVE**:

The main objective of the Apache Spark is Social media websites have emerged as one of the platforms to raise users’ opinions and influence the way any business is commercialized. Opinion of people matters a lot to analyse how the propagation of information impacts the lives in a large-scale network like Twitter. Sentiment analysis of the tweets determine the polarity and inclination of vast population towards specific topic, item or entity. These days, the applications of such analysis can be easily observed during public elections, movie promotions, brand endorsements and many other fields. In this project, we exploited the fast and in memory computation framework 'Apache Spark' to extract live tweets and perform sentiment analysis. The primary aim is to provide a method for analysing sentiment score in noisy twitter streams. This paper reports on the design of a sentiment analysis, extracting vast number of tweets. Results classify user's perception via tweets into positive and negative.

**TECHNOLOGY USED**

Machine Learning Libraries:

● Pandas

● Numpy

● Difflib

● AST

The dataset provided is the **Sentiment140 Dataset**which consists of **1,600,000 tweets** that have been extracted using the Twitter API. The various columns present in this Twitter data are:

* **target:**the polarity of the tweet (positive or negative)
* **ids:**Unique id of the tweet
* **date:**the date of the tweet
* **flag:**It refers to the query. If no such query exists, then it is NO QUERY.
* **user:** It refers to the name of the user that tweeted
* **text:** It refers to the text of the tweet.

***Twitter Sentiment Analysis Dataset: Project Pipeline***

The various steps involved in the **Machine Learning Pipeline** are:

* Import Necessary Dependencies.
* Read and Load the Dataset.
* Exploratory Data Analysis.
* Data Visualization of Target Variables.
* Data Pre-processing.
* Splitting our data into Train and Test sets.
* Transforming Dataset using TF-IDF Vectorizer.
* Function for Model Evaluation.
* Model Building.

**INTRODUCTION:**

As internet is growing bigger, its horizons are becoming wider. Social Media and Micro blogging platforms like Facebook, Twitter, Tumblr dominate in spreading encapsulated news and trending topics across the globe at a rapid pace. A topic becomes trending if more and more users are contributing their opinion and judgements, thereby making it a valuable source of online perception These topics generally intended to spread awareness or to promote public figures, political campaigns during elections, product endorsements and entertainment like movies, award shows. Large organizations and firms take advantage of people's feedback to improve their products and services which further help in enhancing marketing strategies.

One such example can be leaking the pictures of upcoming iPhone to create a hype to extract people's emotions and market the product before its release. Thus, there is a huge potential of discovering and analysing interesting patterns from the infinite social media data for business-driven applications.

* **What is Sentiment analysis?**

Sentiment analysis is the prediction of emotions in a word, sentence or corpus of documents. It is intended to serve as an application to understand the attitudes, opinions and emotions expressed within an online mention. The intention is to gain an overview of the wider public opinion behind certain topics. Precisely, it is a paradigm of categorizing conversations into positive, negative or neutral labels. Many people use social media sites for networking with other people and to stay up-to-date with news and current events. These sites (Twitter, Facebook, Instagram, google+) offer a platform to people to voice their opinions. For example, people quickly post their reviews online as soon as they watch a movie and then start a series of comments to discuss about the acting skills depicted in the movie.

This kind of information forms a basis for people to evaluate, rate about the performance of not only any movie but about other products and to know about whether it will be a success or not. This type of vast information on these sites can used for marketing and social studies. Therefore, sentiment analysis has wide applications and include emotion mining, polarity, classification and influence analysis. Twitter is an online networking site driven by tweets which are 140-character limited messages. Thus, the character limit enforces the use of hashtags for text classification. Currently around 6500 tweets are published per second, which results in approximately 561.6 million tweets per day. These streams of tweets are generally noisy reflecting multi topic, changing attitudes information in unfiltered and unstructured format. Twitter sentiment analysis involves the use of natural language processing to extract, identify to characterize the sentiment content. Twitter sentiment analysis, analyse the sentiment or emotion of tweets. It uses natural language processing and machine learning algorithms to classify tweets automatically as positive, negative, or neutral based on their content. It can be done for individual tweets or a larger dataset related to a particular topic or event.

**TWITTER SENTIMENT ANALYSIS:**

* ***Introduction to Problem:*** Every day massive amount of data is generated by social media users which can be used to analyse their opinion about any event, movie, product or politics. Conventional tools like Apache Storm analyse stream in micro-batch whereas novel tools like Apache Spark process data in real time making analysing and processing of real time data possible.
* ***Platform and Technologies:*** There are different technologies and tools implemented in the project. These are introduced below.

**Apache Spark:**

It is an open source lightning fast cluster computing platform to retrieve streaming data and forwarding to storage system like HDFS, Database Server. It is built on top of Map Reduce and can integrate well with another Apache software. Apache spark is an in memory fast processing system used for large scale data processing. It has come up as an advanced version of Hadoop. Though it implements the MapReduce technology but it processes data even 100 times faster by partitioning on memory and 10 times faster on disk across different nodes. Its structure is based on Resilient Distributed datasets (RDD) which is read only, multi sets of data partitioned and distributed across different node, to ensure fault intolerance and scalability factors. It overcomes the limitation of MapReduce in which data after reducing was stored into a disk by implementing iterative algorithms who fetch data from multiple datasets in a loop thereby implementing repeated database-style querying of data. In this way, the latency involved is reduced thereby making it faster. RDD is basically an abstraction feature which before data processing lays down the execution plan and then later depicts computation using Direct Acyclic Graph (DAG).

The generated DAG acts as a framework to carry out the pattern analysis and processing and task segregation. Further, it has a better edge over other technologies as it is quite easy to implement due to multiple available APIs. Also, the other benefits include high level libraries. This inbuilt feature can deliver support to SQL, machine learning, graph processing and for streaming data. It can access data from different storage sources like HDFS, CASSANDRA, HBase, S3.

**Scala:** It is not only a High Level Functional but also supports Object Oriented Programming language model. This provides it an edge over Java which require more code to be written for the same task as compared to Scala. The major success of Scala is that Apache Spark is itself implemented in Scala. There are vast number of packages available in Scala language for Apache Spark. Thus, we proceeded with implementation in Scala as compared to Python or Java.

**Twitter:** It is an online social media platform which is suitable for our use case due to number of factors. Firstly, the amount of relevant data is much larger for twitter as compared to blogs or review websites. Secondly, response on twitter is general and prompt. Other social media giants like Facebook does not provide much data so using their public API was not considered. Finally, most twitter users voice their opinion about other people like actors, products: in case they bought a new phone or unsatisfied with customer service behaviour as opposed to other social media where users post most status and pictures of themselves. These factors make twitter a logical choice for real time data analysis.

**IntelliJ Idea:** It is an Integrated Development Environment to build, run and test code. It is closed source but community edition of the software is provided free of cost. It offers support for SBT plugin which is used to import Apache Spark dependencies and build the project. Intellij Idea professional edition is used along with SBT plugin which is a build tool, an alternative for maven build tool. SBT makes it easy to define dependencies and import libraries and dependencies.

**Why should we use Sentiment Analysis?**

* **Invaluable Marketing:** Using sentiment analysis companies and product owners use can use sentiment analysis to know the demand and supply of their products

through comments and feedback from the customers.

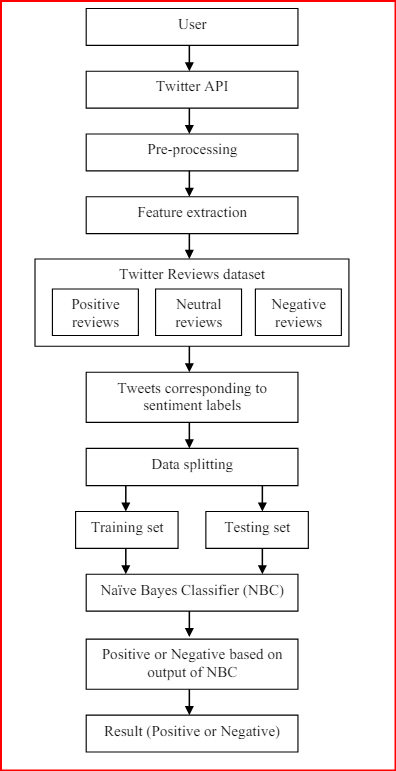
* **Identifying key emotional triggers:** In psychology and other medical treatment institutions, sentiment analysis can be used to detect whether the individuals’ emotion is normal or abnormal, and based on the data record they can decide person health.
* **Politics:** In the political field, candidates to be elected can use sentiment analysis to predict their political status, to measure people’s acceptance. It can also be used to predict election results for electoral board commissions.
* **Education:** Universities and other higher institutes like colleges can use sentiment analysis to know their student’s feedback and comment, therefore they can take consideration to revise or improve their education curriculum.

**SENTIMENT CLASSIFICATION ANALYSIS USING MACHINE LEARNING:**

The Figure demonstrates the block diagram of Sentiment Classification Analysis of Tweets on Twitter Data Using Machine Learning Algorithm. They examine the tweets that users had uploaded with hashtags to indicate their perspectives. Subscribe to the Twitter API, authenticating it with the help of the access token, access secret, consumer key, and consumer secret, and then start collecting data from Twitter. Use the R packages Twitter and Roath to accomplish. Once the keys have been created, gathering information on the desired person, product, etc. is simple. Multiple misspellings, symbols, Uniform Resource Locator (URL), and hashtags can be found in the twitter data that has been collected. These statistics produce substandard outcomes. The pre-processing phase is required to obtain correct findings in order to prevent this. All link tags, punctuation marks, hash tags, special characters, emoticons etc. are removed during pre-processing. The tweets are grammatically corrected if necessary. Stop words that don't modify the tweets are eliminated.

**Removal**: For the purposes of processing, user names and URLs included in the data are irrelevant. Therefore, most identities and URLs are either deleted or changed to standard tags.

**Stemming**: This refers to the process of reducing multiple types of words with the same or comparable meanings by replacing them with their roots. To make feature extraction easier, the included procedures must strive to make the data machine readable.



**Fig: Block Diagram of Sentiment Analysis**

**BLOCK DIAGRAM OF SENTIMENT ANALYSIS USING MACHINE LEARNING:**

They need to extract relevant features for sentiment analysis once the tweets have been cleaned. The quantity and quality of features have an impact on the output that a model produces. The dataset’s aspect (adjective) is extracted using this procedure. This component is subsequently used to demonstrate positive and negative polarity in a statement, supporting in the classification of public opinion. Now that the data has been cleaned, the features are extracted to create significance data, which in turn creates a trained data set that includes both positive and negative data.Following feature extraction, this Text Blob approach is used to analyse the dataset to determine the sentiment ratings. Text Blob outperforms the original dataset according to noise removal. A performance evaluation of the original and modified data using the Text Blob sentiment scores. A Python package called Text Blob is used for Natural Language Processing (NLP) activities like component tagging, sentiment classification, extracting noun phrases, translations, classifications, and more. To extract the emotions from Twitter tweets, we use Text Blob. Every tweet receives two properties from the Text Blob sentiment function: a polarization score between and an objectivity score between.

Positives, neutrality, and negatives polarization scores, correspondingly, represent positive, unbiased, and negative expressions. After determining sentiment scores, the dataset is separated into a training set and a testing set using an 80:20 ratio, meaning that 80% of the original data is used for training and 20% for testing. One Machine Learning algorithm that makes advantage of the Bayes algorithm and the strong independence among the features is the Naive Bayes classifier. A significant amount of information is typically utilized with a Naive Bayes algorithm because it is so simple to construct. It is utilized as one of the fundamental methods for classifying texts. Finding the positive, unbiased, and negative sentiments of messages taken from Twitter is done using sentiment analysis and the Naive Bayes classifier.

**RESULT ANALYSIS:**

People use Twitter more frequently than any other site to share their opinions and feelings through tweets. Subscribe to the Twitter API (Application Programming Interface), authenticating it with the help of the access token, access secret, consumer key, and consumer secret, and then start collecting data from Twitter. 2000 tweets were used to train the algorithms at first. This model is used with given pre-processed data after training. A total of 260 tweets were deleted out of 3000, of which 2680 are actual negatives and 160 are true positives. Training data is requiring the 80% of original data and testing data requires the 20% of original data. They have utilized four performance standards to assess the effectiveness of the Machine Learning model of the Naive Bayes classifier: accuracy, precision, recall, and F1-score.

**Accuracy:** The overall correctness of the categorization produced is connected to the accuracy metric. The ratio of cases that were successfully categorized to all instances is known as accuracy. **𝐴𝑐𝑐𝑢𝑟𝑎𝑐𝑦 = 𝑇𝑃 + 𝑇𝑁 𝑇𝑃 + 𝑇𝑁 + 𝐹𝑁 + 𝐹𝑃.**

**Precision**: The precision measure is the percentage of tweets successfully classified for the specified sentiment terms out of all tweets correctly categorized for this sentiment.

**𝑃𝑟𝑒𝑐𝑖𝑠𝑖𝑜𝑛 = 𝑇𝑃 𝑇𝑃 + 𝐹𝑃.**

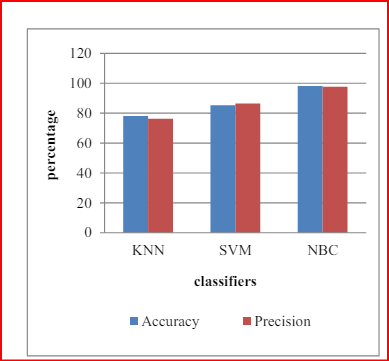
**F1-Score:** The weighted average of Precision and Recall is known as the F1-Score. Therefore, F1-Score accounts for both false positives and false negatives. The F1score is mostly valuable than the accuracy even though it is not as simple to read, particularly when the class distribution is unequal. The most effective cases for accuracy are those in which the cost of false positives and false negatives is similar. **𝐹1 − 𝑆𝑐𝑜𝑟𝑒 = 2 ∗ (𝑅𝑒𝑐𝑎𝑙𝑙 ∗ 𝑃𝑟𝑒𝑐𝑖𝑠𝑖𝑜𝑛) (𝑅𝑒𝑐𝑎𝑙𝑙 + 𝑃𝑟𝑒𝑐𝑖𝑠𝑖𝑜𝑛).**

Where, True Positive (TP) Positive observations and evaluations of reviews are produced through classification. True Negative (TN) Reviews that have been seen are not positive and are rated as one and the same. False Positive (FP) Reviews were observed, however they were evaluated positively. False Negative (FN) Identified reviews show that they are favourable, but the classifier rates them as undesirable. The performance analysis of Sentiment Classification Analysis of Tweets using Naive Bayes Classifier (NBC) is compared with other classification models as Nearest Neighbour (KNN) and Support Vector Machine (SVM) represented in below Table.

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameters** | **KNN** | **SVM** | **NBC** |
| Accuracy | 78.2 | 85.3 | 98.1 |
| Precision | 76.3 | 86.6 | 97.6 |
| Recall | 77.8 | 84.9 | 96.4 |
| F1-Score | 79.1 | 87.4 | 98.5 |

**Table 1:** PERFORMANCE OF DIFFERENT CLASSIFIERS BASED SENTIMENT CLASSIFICATION

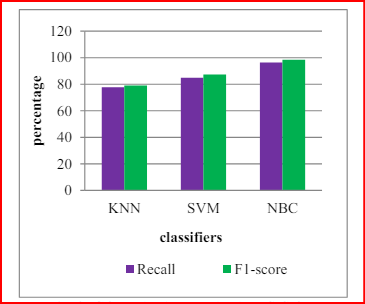
Following shows a graphical representation of accuracy and precision.



**Fig:** COMPARATIVE ANALYSIS OF DIFFERENT CLASSIFIERS ACCURACY

AND PRECISION PARAMETERS

Following shows a graphical representation of recall and the F1 score.

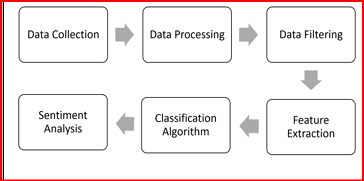


**Fig:** COMPARATIVE ANALYSIS OF DIFFERENT CLASSIFIERS RECALL AND

F1- SCORE PARAMETERS.

**CASE STUDY:**

Project involves the usage of Apache Spark to analyse real time tweets. The objective of our case study is to find the polarity of the words (in tweets) retrieved.

****

**Fig:** Framework for Twitter Analysis.

Each step in the framework involves several sub-tasks.

1. **Data collection:** Data in the form of raw tweets is retrieved by using the Scala library “Twitter4j” which provides a package for real time twitter streaming API. The API requires us to register a developer account with Twitter and fill in parameters such as consumer Key, consumer Secret, access Token access, and Token Secret. This API allows to get all random tweets or filter data by using keywords. Filters supports to retrieve tweets which match a specific criterion defined by the developer. We used this to retrieve tweets related to specific keywords which are taken as input from users. Initially, we set at least set an application name and mode. We execute the program in local mode instead of cluster. Then, input array of keywords is provided as an argument to Streaming Context “ssc” using “sc” where “sc” is spark context.
2. **Data Processing:**  Data processing involves Tokenization which is the process of splitting the tweets into individual words called tokens. Tokens can be split using whitespace or punctuation characters. It can be unigram or bigram depending on the classification model used. The bag-of words model is one of the most extensively used model for classification. It is based on the fact of assuming text to be classified as a bag or collection of individual words with no link or interdependence.

The simplest way to incorporate this model in our project is by using unigrams as features. It is just a collection of individual words in the text to be classified, so, we split each tweet using whitespace. For example, the tweet “Met aziz today !!” is split from each whitespace as follows. {Met Aziz !!”} The next step in data processing is normalization by conversion of tweet into lowercase. Tweets are normalized by converting it to lowercase which makes its comparison with a dictionary easier.

1. **Data Filtering:** A tweet acquired after data processing still has a portion of raw information in it which we may or may not find useful for our application. Thus, these tweets are further filtered by removing stop words, numbers and punctuations. Stop words: For example, tweets contain stop words which are extremely common words like “is”, “am”, “are” and holds no additional information. These words serve no purpose and this feature is implemented using a list stored in stopfile.dat.
2. **Feature Extraction**: TF-IDF is a feature vectorization method used in text mining to find the importance of a term to a document in the corpus. Feature extraction involves “mlib” library of Apache Spark. The recommended API is the Data Frame based API. This feature is useful for a case where we need to find trending topics or to create word clouds. However, this project is more focused towards finding sentiment in twitter streams so TF-IDF is not implemented.
3. **Sentiment Analysis:** Sentiment analysis is done by using custom algorithm which finds polarity as below. Finding polarity: For discovering the polarity, we used a simple algorithm of counting positive and negative words in a tweet. For both, positive and negative words, different lists were made. Next step is to compare every word in a tweet against both these lists. If the current word matches a word in positive list, then a score of 1 is incremented and if a negative word is found then it is decremented. More positive words lead to higher sentiment. However, Sandford NLP can be used to predict accurate sentiment analysis which provide complex algorithms to predict it.
4. **Sentiment Analysis output:** The output contains a list of tweets in real time along with their sentiment score on the left-hand side. The first tweet has score of -2 which is due to two negative keywords. Next two tweets are positive as they contain keywords like “good” and “great. Both these words are in the positive words list. It is to be noted that if a tweet has a score of 0, then it is ignored from final output. The problem with neutral tweets is that they serve no purpose as they don’t convey any sentiment towards the product.

**CODE:**

import numpy as np

import pandas as pd

import time

import matplotlib. pyplot as plt

from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator

!ls ../input/data-science-tweets/tweets/

t1 = time.time()

df = pd.read\_csv('../input/data-science-tweets/tweets/data\_science.csv')

t2 = time.time()

print('Elapsed time [s]: ', np.round(t2-t1,2))

df.date = pd.to\_datetime(df.date)

df['year'] = df.date.dt.year

df['month'] = df.date.dt.month

df.info()

df.year.value\_counts().sort\_index().plot(kind='bar')

plt.title('Year of Tweet')

plt.grid()

plt.show()

df.month.value\_counts().sort\_index().plot(kind='bar')

plt.title('Month of Tweet')

plt.grid()

plt.show()

df.language.value\_counts()

df.tweet

text = " ".join(txt for txt **in** df.tweet)

stopwords = set(STOPWORDS)

stopwords = stopwords.union(set(['https', 't', 'co', '&amp;', 'amp']));

n\_words = 250

wordcloud = WordCloud(stopwords=stopwords, max\_font\_size=50, max\_words=n\_words,

width = 600, height = 400,

collocations=False,

background\_color='white').generate(text)

plt.figure(figsize=(14,10))

plt.imshow(wordcloud, interpolation='bilinear')

plt.axis('off')

plt.show()

for y **in** range(2010,2021+1):

df\_temp = df[df.year==y]

print('Year =',y, ':')

text = ' '.join(txt for txt **in** df\_temp.tweet)

wordcloud = WordCloud(stopwords=stopwords, max\_font\_size=50, max\_words=n\_words,

width = 600, height = 400,

collocations=False,

background\_color='white').generate(text)

plt.figure(figsize=(14,10))

plt.imshow(wordcloud, interpolation='bilinear')

plt.axis('off')

plt.show()

**Explanation of Code:**

This code is used to analyzed data science tweets from 2010 to 2020. It imports the necessary libraries, reads the data from a CSV file, and then performs the following steps:

* It extracts the date and year from each tweet.
* It counts the number of tweets per year and month.
* It creates a word cloud of the most common words used in the tweets.
* It generates a word cloud for each year from 2010 to 2020.

Here is a detailed explanation of each step:

**import numpy as np**

**import pandas as pd**

**import time**

**import matplotlib. pyplot as plt**

from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator

This code imports the necessary libraries. The numpy library is used for numerical operations, the pandas library is used for data manipulation and analysis, the time library is used for measuring time, the matplotlib. pyplot library is used for plotting, and the wordcloud library is used for generating word clouds.

**!ls ../input/data-science-tweets/tweets/**

This code lists the contents of the ../input/data-science-tweets/tweets/ directory. This directory contains the CSV file that contains the data science tweets.

**t1 = time.time()**

**df = pd.read\_csv('../input/data-science-tweets/tweets/data\_science.csv')**

**t2 = time.time()**

**print('Elapsed time [s]: ', np.round(t2-t1,2))**

This code reads the CSV file into a Pandas DataFrame. The time.time() function is used to measure the time it takes to read the file. The np.round() function is used to round the time to two decimal places.

**df.date = pd.to\_datetime(df.date)**

**df['year'] = df.date.dt.year**

**df['month'] = df.date.dt.month**

This code extracts the date and year from each tweet. The pd.to\_datetime() function is used to convert the date column to a datetime object. The dt.year and dt.month attributes are used to extract the year and month from the datetime object.

**df.info() :**This code prints a summary of the DataFrame. This summary includes the number of rows and columns in the DataFrame, the data types of each column, and the number of non-null values in each column.

**df.year.value\_counts().sort\_index().plot(kind='bar')**

**plt.title('Year of Tweet')**

**plt.grid()**

**plt.show()**

This code counts the number of tweets per year and plots the results as a bar chart. The value counts () function is used to count the number of occurrences of each unique value in the year column. The sort index () function is used to sort the results in ascending order. The plot(kind='bar') function is used to plot the results as a bar chart. The title () function is used to set the title of the plot. The grid () function is used to add a grid to the plot. The show () function is used to display the plot.

**df.month.value\_counts().sort\_index().plot(kind='bar')**

**plt.title('Month of Tweet')**

**plt.grid()**

**plt.show()**

This code counts the number of tweets per month and plots the results as a bar chart. This code is similar to the code used to plot the number of tweets per year.

**df.language.value\_counts()**

This code counts the number of tweets in each language. The value counts () function is used to count the number of occurrences of each unique value in the language column.

**df.tweet\_text = " ".join(txt for txt in df.tweet)**

This code creates a new column called tweet text that contains the text of all of the tweets. The join () function is used to join the text of all of the tweets into a single string.

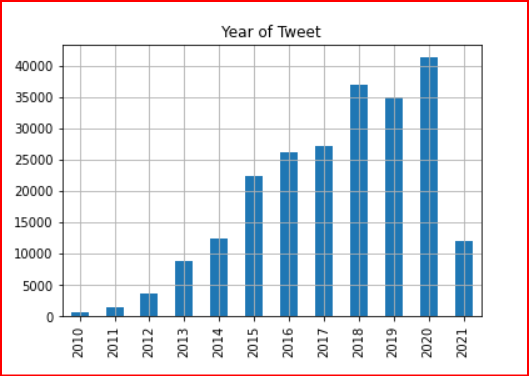
**stopwords = set(STOPWORDS)**

**stopwords = stopwords. union(set(['https', 't', 'co', '&amp;', 'amp']));**

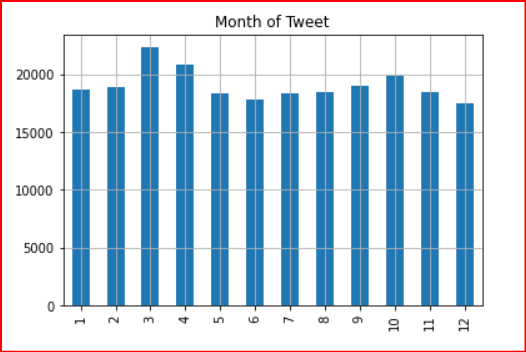
This code creates a set of stop words. Stop words are words that are common in a language but do not add much meaning to a code.

**OUTPUT:**

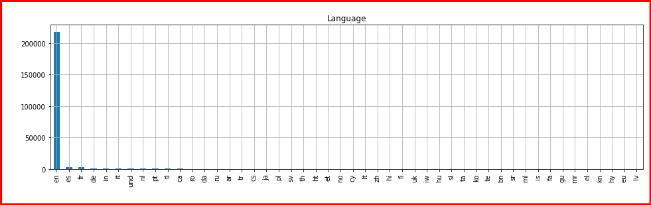
Year of Tweet:

****

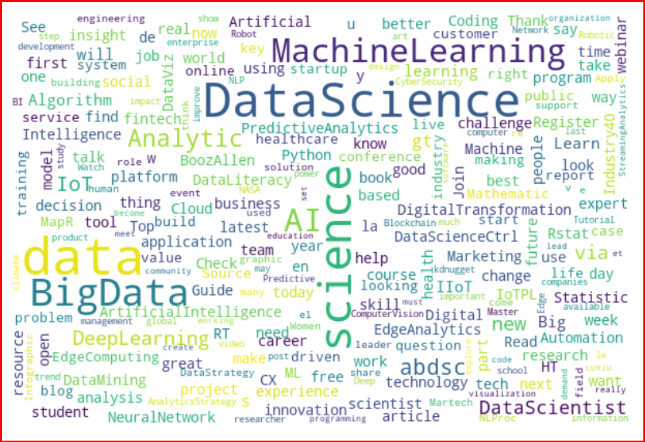
Month of Tweet:

****

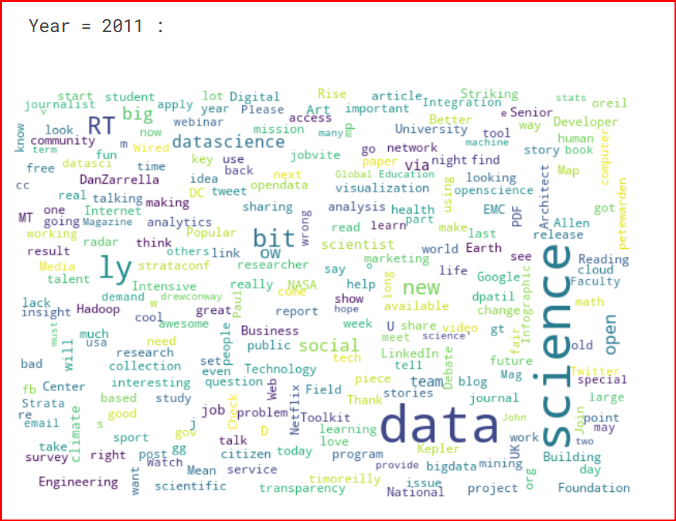
Language Output:

****

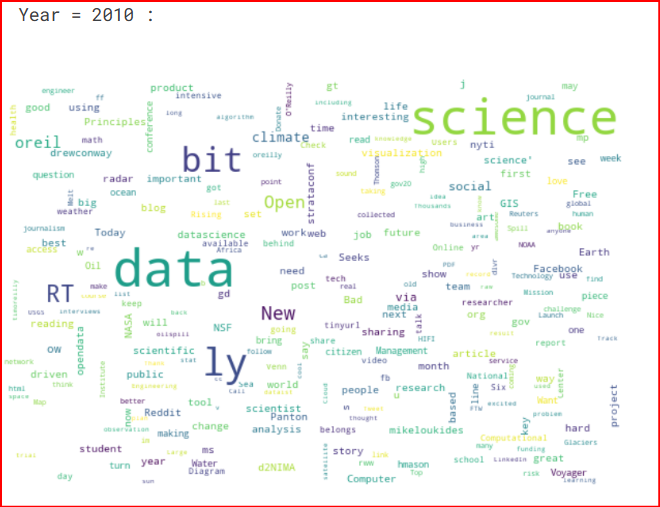
Yearly Output:

****

Year 2011:

****

Year 2010:

****

**CONCLUSION:**

In this approach, Sentiment Classification Analysis of Tweets on Twitter Data using Machine Learning algorithm is described. In order to decide which features are best. For training and testing word features as well as determining the sentiment polarity of each tweet, Naive Bayes Classifier (NBC) is utilized. Sentiment analysis is a new and quickly growing area of the decision-making process. The project’s purpose is to evaluate the sentiments on a topic that are taken from Twitter and determine whether they are good, negative, or neutral. Information obtained from the Twitter API by approved users.

The Text Blob approach was used on the information to find sentiment ratings after feature extraction. Comparative performance analyses of different classifiers are described in result analysis phase in terms of accuracy, precision, recall and F1- score. Obtained performance parameters values as Accuracy 98.1%, Precision 97.6%, Recall 96.4% and F1-Score 98.5%. Future study should include the addition of extra features, which will raise prediction accuracy.