LITERATURE REVIEW:  
1. EXISTING SYSTEMS:

1. Vader Sentiment Analysis

Hutto and Gilbert created the lexicon- and rule-based sentiment analysis tool known as Vader Sentiment Analysis. It is made specifically to evaluate the tone of tweets and other social media messages. The programme is based on a corpus of sentiment-related words and expressions, each of which is given a score indicating its positive or negative polarity.

To further hone its analysis, Vader applies a set of grammatical and syntactical principles. For instance, it can determine a word or phrase's emotional intensity based on its use of capitalization, punctuation, and emphasis. Vader is also capable of handling emoticons, context-sensitive sentiment, and negation.

Vader Sentiment Analysis produces a sentiment score, which ranges from -1 to +1, with -1 representing the most negative emotion, 0 representing neutral sentiment, and +1 representing the highest positive sentiment. Additionally, Vader offers distinct ratings for sentiment that is favourable, negative, or neutral.

In general, Twitter data can be utilised to analyse sentiment using Vader Sentiment Analysis, which is a popular and useful technique. Its reliance on a predefined language and the possibility for prejudice and inaccuracy in specific situations are some of its drawbacks. Vader should therefore be used in conjunction with other sentiment analysis methods, and its results should be carefully considered in light of the particular context.

1. TEXTBLOB

Python's TextBlob package can be used to process textual data, including twitter sentiment analysis. It offers a straightforward and understandable interface for carrying out various natural language processing (NLP) operations and is developed on top of the Natural Language Toolkit (NLTK).

The power of TextBlob to perform sentiment analysis is one of its primary characteristics. A sentiment classifier is trained using a machine learning algorithm on a sizable dataset of movie reviews. The sentiment of new text, like tweets, can then be classified using the classifier.

Two values are returned by TextBlob's sentiment analysis function: a polarity score and a subjectivity score. The polarity score has a range of -1 to +1, with -1 being the most negative feeling, 0 representing neutral attitude, and +1 representing the most positive. A statement is given a subjectivity score between 0 and 1, with a score of 0 denoting objectivity and a score of 1 denoting extreme subjectivity.

Other helpful functions offered by TextBlob for processing textual data include language translation, noun phrase extraction, and parts-of-speech tagging. It is a popular option for various NLP applications, including sentiment analysis of Twitter data, due to its straightforward and user-friendly API.

It is important to keep in mind that TextBlob's sentiment analysis algorithm has significant drawbacks, including its reliance on a pre-defined dataset of movie reviews that might not be indicative of the sentiments conveyed in tweets. Additionally, TextBlob might not work as well with non-English texts or texts that use figurative language like sarcasm or irony. Thus, it is crucial to combine the results of TextBlob's sentiment analysis with those from other methods and to give careful thought to the particular context and objectives of the analysis.

1. IBM WATSON

IBM IBM created a collection of artificial intelligence (AI) technologies and services called Watson. Watson has the ability to perform sentiment analysis, which can be used with a variety of textual data types, including tweets.

Natural language processing and machine learning algorithms provide the foundation of Watson's sentiment analysis feature. It classifies the sentiment of new text using a pre-trained model that was built on a sizable corpus of text data. Idioms, sarcasm, and irony are just a few of the sophisticated linguistic patterns that Watson's sentiment analysis is capable of understanding.

The sentiment score provided by Watson's sentiment analysis function ranges from -1 to +1, with -1 denoting the most negative sentiment, 0 denoting neutral sentiment, and +1 denoting the most positive sentiment. Additionally, Watson offers distinct scores for each type of sentiment—positive, negative, and neutral—as well as an overall confidence score that shows the algorithm's level of assurance in its ability to classify sentiment.

The capability of Watson to analyse vast quantities of text input accurately and swiftly is one advantage of employing Watson for sentiment analysis. It may be integrated into a variety of apps and work processes and analyse text in several languages. Other NLP capabilities offered by Watson include entity recognition, concept extraction, and natural language processing.

Watson's sentiment analysis feature does, however, have significant drawbacks, including its reliance on pre-defined models and the potential for bias and inaccuracy in specific situations. As a result, it's crucial to combine Watson's sentiment analysis findings with other methods and to carefully examine the particular environment and study aims.

1. GOOGLE CLOUD NLP API

Developers can use the Google Cloud Natural Language API, a service offered by Google Cloud Platform, to perform natural language processing (NLP) operations on text such as sentiment analysis, entity recognition, syntax analysis, and others. It makes use of machine learning algorithms to comprehend the structure and meaning of text data and offers perceptions into the tone, subjects, and ideas covered in the text.

A range of text inputs, such as social network posts, customer reviews, news articles, and more, can be analysed using the Natural Language API. The text's language can be recognised, things like persons, places, and organisations can be extracted, and sentiment analysis can be used to assess if the text is good, negative, or neutral. Additionally, it is capable of parsing sentence structures, identifying speech sounds, and extracting significant phrases through syntax analysis.

The RESTful interface of the API makes it simple for developers to include it into their applications. Java, Python, and Go are just a few of the programming languages that are supported. The Google Cloud Console is another tool that developers may use to interact with the API and learn more about its features.

For companies and developers that want to quickly and effectively glean insights from enormous volumes of text data, the Google Cloud Natural Language API is a potent tool. Businesses can use it to increase customer happiness, evaluate brand reputation, and better understand the wants and expectations of their target market.

1. HOOTSUITE

A social media management software called Hootsuite offers Hootsuite Insights, a social media listening and analytics tool. It gives companies and organisations the ability to keep an eye on social media platforms, follow conversations, and examine social media metrics to learn more about consumer mood, market trends, and competition activity.

Users may design unique dashboards using Hootsuite Insights that show real-time statistics on important metrics like engagement rates, share of voice, and sentiment analysis.

Additionally, they can keep watch on particular hashtags, keywords, or mentions on social media platforms and get notifications when significant discussions or trends start.

Additionally, Hootsuite Insights provides advanced analytics tools including content analysis, influencer identification, and demographic research. Users may determine which audiences are the most active, monitor the effectiveness of social media marketing, and assess how social media usage affects corporate goals.

A number of social networking platforms, including Facebook, Twitter, Instagram, and LinkedIn, as well as external data sources like Google Analytics, are integrated with the platform. Users may share findings with stakeholders in an understandable way thanks to Hootsuite findings' user-friendly interface and customised reporting options.

In general, Hootsuite Insights is an effective tool for companies and organisations seeking actionable

PROPOSED MODEL

A dataset is a collection of data that is structured and organized in a specific way for analysis purposes. Our dataset includes multiple rows and columns where in rows are the single observation points or data points and column represents the variable or attribute of that observation.

In data science project, quality of data is very important because that directly affects the accuracy and analysis of the results drawn from the data.

## DATASET CREATION

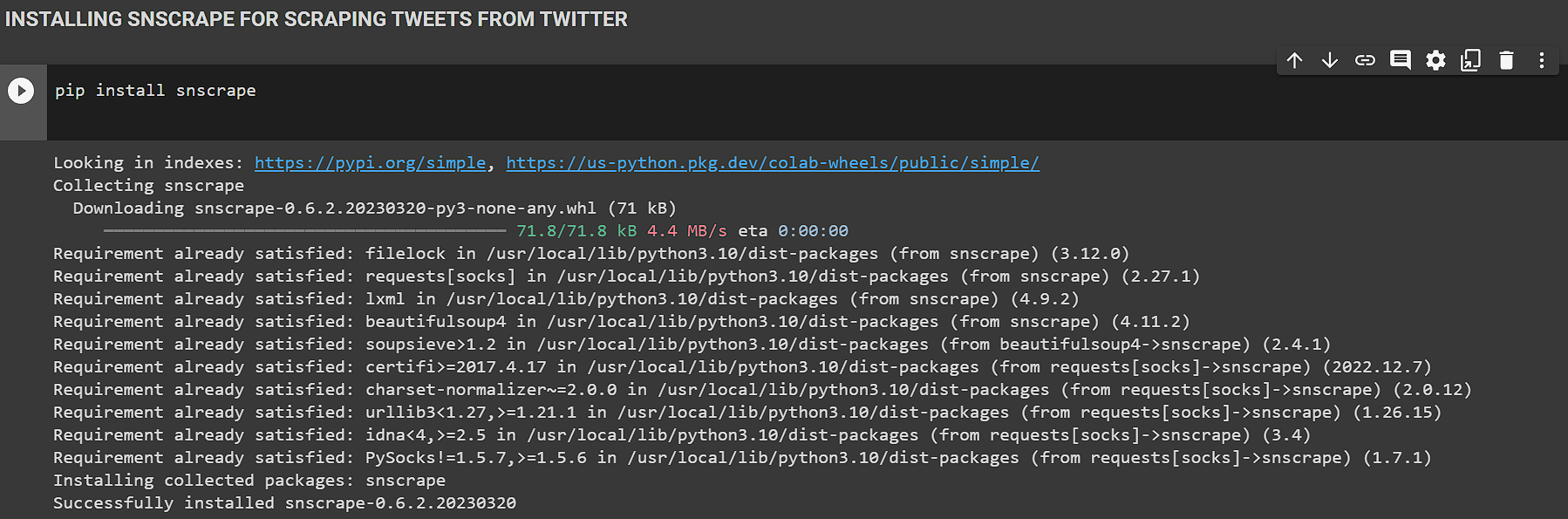
In this investigation, data is scraped from twitter using the Snscrape module which eases us to scrape any amount of data without the use of API key authentication of twitter and do not have the limiting condition of 3200 tweets of Twitter API.

Our dataset is on the recent Adani Vs Hindenburg Fiasco. We will fetch tweets in a date and time frame and those will total to 20,000 tweets in our dataset.

We install necessary dependencies and scrape the data using the below query command from 2023-02-01 to 2023-03-09 date and time respectively.

Our data in this paper is scraped from Twitter with the query= “Adani OR Hindenburg OR #AdnaiVsHindenburg OR #HindenburgReport.”

PIP INSTALLING SNSCRAPE



INSTALLING LIBRARIES:

1. PANDAS

Pandas is a Python library for analyzing and manipulating data. For working with structured data, it offers a potent set of capabilities, including data frames, series, and robust data selection and manipulation features. Machine learning and data science both make extensive use of Pandas.

1. SNSCRAPE.MODULES. TWITTER

Snscrape. Modules. Python is a twitter module is used to scrape information from Twitter. For collecting tweets, user profiles, and other Twitter data, it offers a simple interface. It can be used for many different things, including sentiment analysis and monitoring social media.

1. NUMPY

The Python package NumPy is used for data analysis and scientific computing. It offers a strong array computing architecture that makes it possible to create and work with sizable multi-dimensional arrays and matrices. A lot of people use NumPy.

1. MATPLOTLIB

The Python module Matplotlib is used for data visualization. It offers a variety of tools for generating graphs, charts, and other data visualizations. Because of its flexibility, Matplotlib can be used for a variety of tasks, from exploratory data analysis to producing figures suitable for publishing.

1. SEABORN

The Python module Seaborn is used to visualize statistical data. It offers a higher-level interface for developing complex statistical visualizations on top of Matplotlib. For the purpose of displaying distributions, regressions, and categorical data, Seaborn offers a variety of plotting functions.

1. NLTK

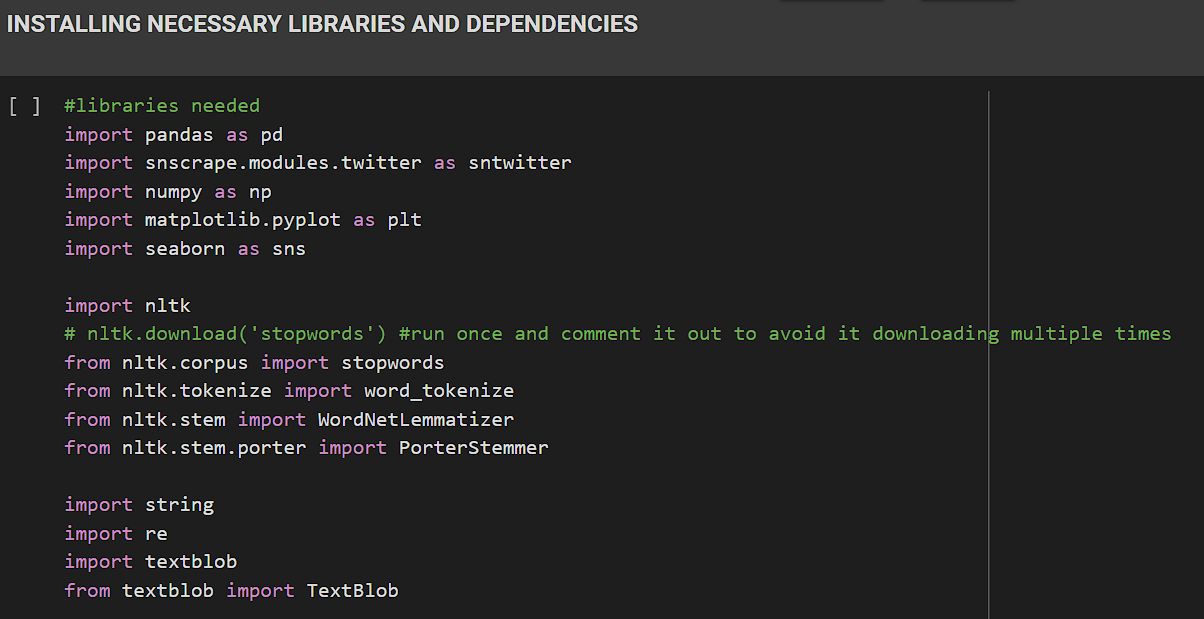
The Python module NLTK (Natural Language Toolkit) is used to process natural language. For processing and analyzing text data, it offers a variety of techniques and resources, including tokenization, stemming, lemmatization, part-of-speech tagging, and others.

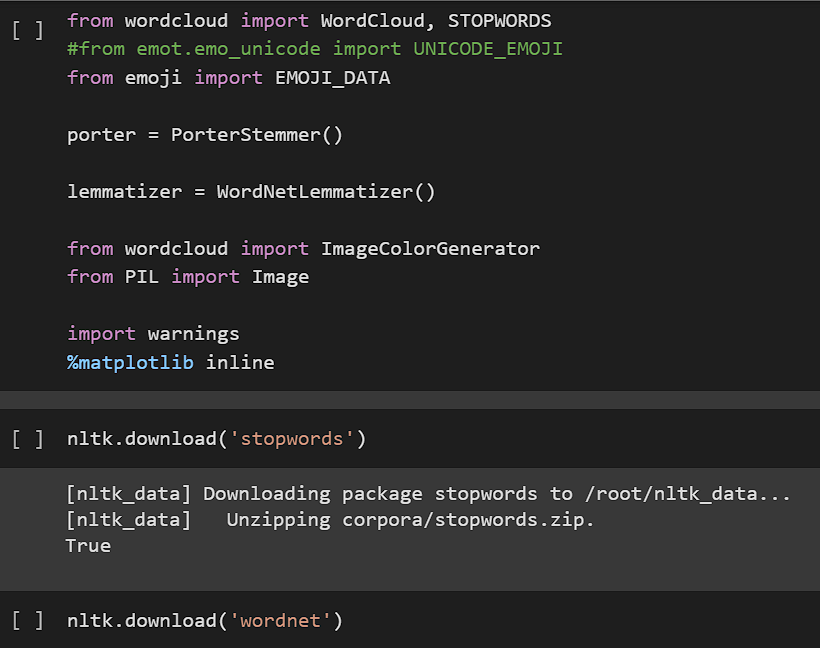
1. STRING

The Python library's string module offers several functions for working with strings. It offers several string manipulation operations, including text formatting, searching, and replacement.

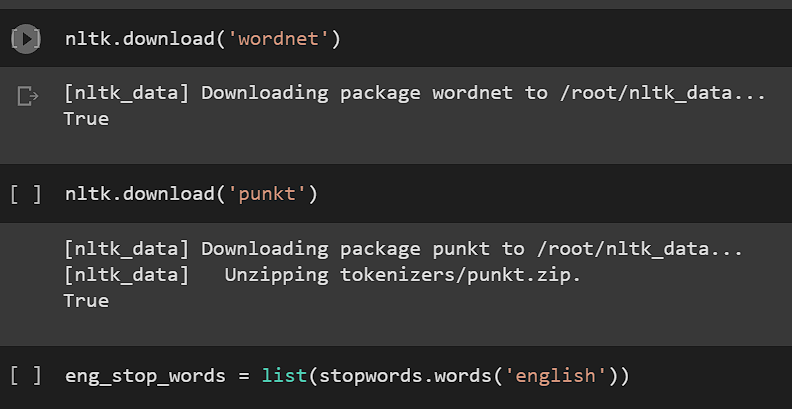
1. TEXTBLOB

The Python package TextBlob is used for processing natural language. It offers a high-level user interface for standard NLP tasks including sentiment analysis, part-of-speech tagging, and noun phrase extraction. TextBlob, which is based on NLTK, offers a streamlined user interface for typical natural language processing tasks.



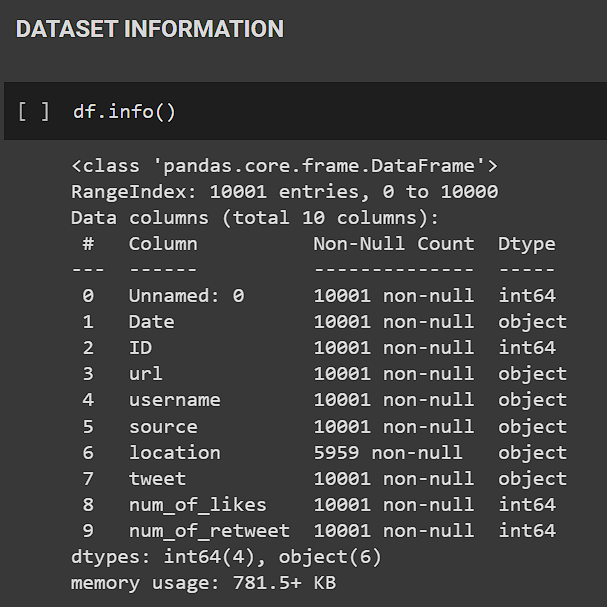


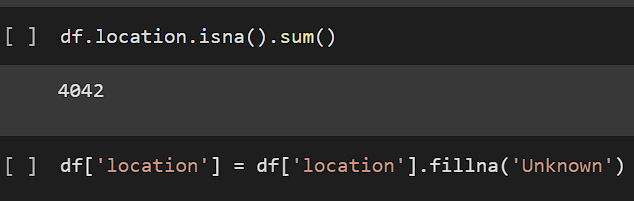
**Stating English stop words for our dataset**



WRITING QUERY FOR SCRAPING DATA FROM TWITTER:







DATASET ATTRIBUTE DESCRIPTION:

1. DATE: Date of tweet creation
2. ID: Unique Id no. of the user
3. URL: The link to the Twitter account of user
4. USERNAME: Name of user
5. SOURCE: Source of tweet creation (Android/Desktop)
6. LOCATION: User location at time of tweet
7. TWEET: Actual tweet content
8. NUM\_OF\_LIKES: Likes on the tweet.
9. NUM\_OF\_RETWEETS: Retweets on the tweets

These fields will later help us in making an interactive dashboard for this Twitter dataset for easy visualization.

TABLE II Statistics of Dataset Used

|  |  |  |  |
| --- | --- | --- | --- |
| **DATASET** | **POSITIVE** | **NEGATIVE** | **NEUTRAL** |
| SCRAPED | 2536 | 1697 | 5768 |

## Pre-Processing of Tweets

We are fetching the data from Twitter social media platform since it is predominantly consisting of text data and we can easily employ NLP and text processing technique to it.

Twitter limits us to publish only 140 characters of content at a time in a tweet. It also has a lot of slang words, hashtags, @username handle mentions which creates a problem in processing of data. Hence pre-processing of data is very crucial here.

* STEP1: REMOVING @USER TWITTER HANDLES

We could see that @user and URL of the tweets are of no use in the sentiment analysis of our tweets. So, we simply remove them by a pattern matching function in python.

Regular expressions and the Python re module can be used to remove @user Twitter handles from a string.

We begin by importing the regular expressions module, re. Next, we define a sample tweet example string that includes a few @user Twitter handles. We replace every instance of a pattern that matches @user handles with an empty string using the re.sub() function. Any string that begins with "@" and is followed by one or more non-whitespace characters is compatible with the regular expression pattern @[s]+. The cleaned tweet is then printed sans the @user handles.

* STEP2: REMOVING PUNCTUATION, NUMBERS, SPECIAL CHARACTERS

We remove the full stop(.), exclamation mark(!) and other punctuations from our tweet data so that it can be processed. Also remove any number and special characters from the tweet which do not contribute to the sentiment analysis.

With the use of regular expressions and the re module, you may eliminate punctuation, digits, and special characters from a string in Python.

We begin by importing the regular expressions module, re. We then define a string text example that includes punctuation, numbers, and special characters. We replace every instance of a pattern matching any non-word character or whitespace character with an empty string by using the re.sub() function. Any character that is not a word character or a whitespace character fits the regular expression pattern [ws]. Additionally, we replace every instance of a pattern that matches one or more numbers with an empty string using the re.sub() function. One or more digits can be found that match the regular expression pattern d+. The cleaned text is then printed without

* STEP3: REMOVE STOP WORDS

Remove stop words from the tweet dataset as they provide no sense to the sentiment mining in our research.

For Ex: “This issue is for general public” herein the stop words are this, is, for which are to be removed and the final processed text is issue general public.

In Python, you can use the nltk (Natural Language Toolkit) package to get rid of stop words from a string.

Using the nltk.download() function, we first import the nltk library before downloading the stop words corpus. Then, a sample string text with a few stop words is defined. Using the split() method, we divided the text into words. In order to obtain a list of English stop words, we utilise the stopwords.words('english') function. Using a list comprehension, we go over the words repeatedly, keeping just those that aren't stop words. Using the join() technique, we combine the clean words back into a string. The cleaned text is then printed without any stop words.

Using pip, install the NLTK library Run nltk.download('stopwords') in your Python environment to install nltk and download the stop words corpus.

* STEP4: TOKENIZATION

Tokens are the individual words of the string text and we create tokens of the cleaned text and this process is Tokenization. We use the NLTK package of python.

The process of separating a string into separate words or tokens is known as tokenization. The Python nltk (Natural Language Toolkit) library may tokenize a string.

To download the punkt data for tokenization, we first import the nltk library and use the nltk.download() function. Next, we create a tokenization example string text. The string is tokenized into words using nltk.word\_tokenize(). The tokens are then printed as a list of strings.

The nltk.word\_tokenize() method separates words based on whitespace and punctuation to tokenize the input text. Additionally, it correctly handles words with hyphens and contractions. Use the nltk.sent\_tokenize() function instead if you wish to tokenize a text into sentences.

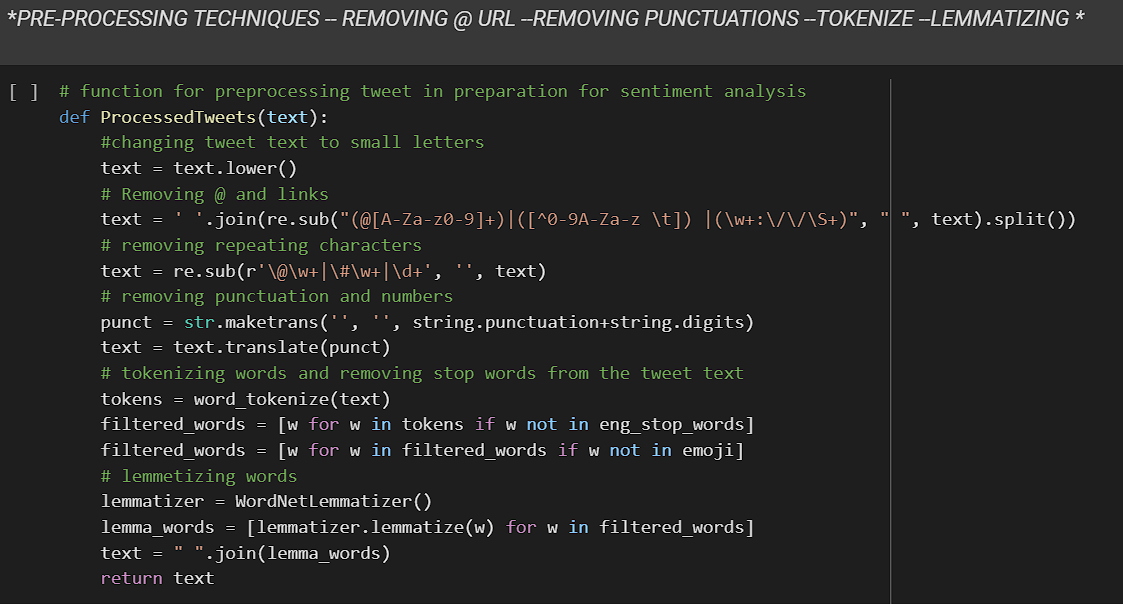
* STEP5: STEMMING

Stemming is the process of scrapping the suffixes of the word {“ing”, “ly”, “es”}.

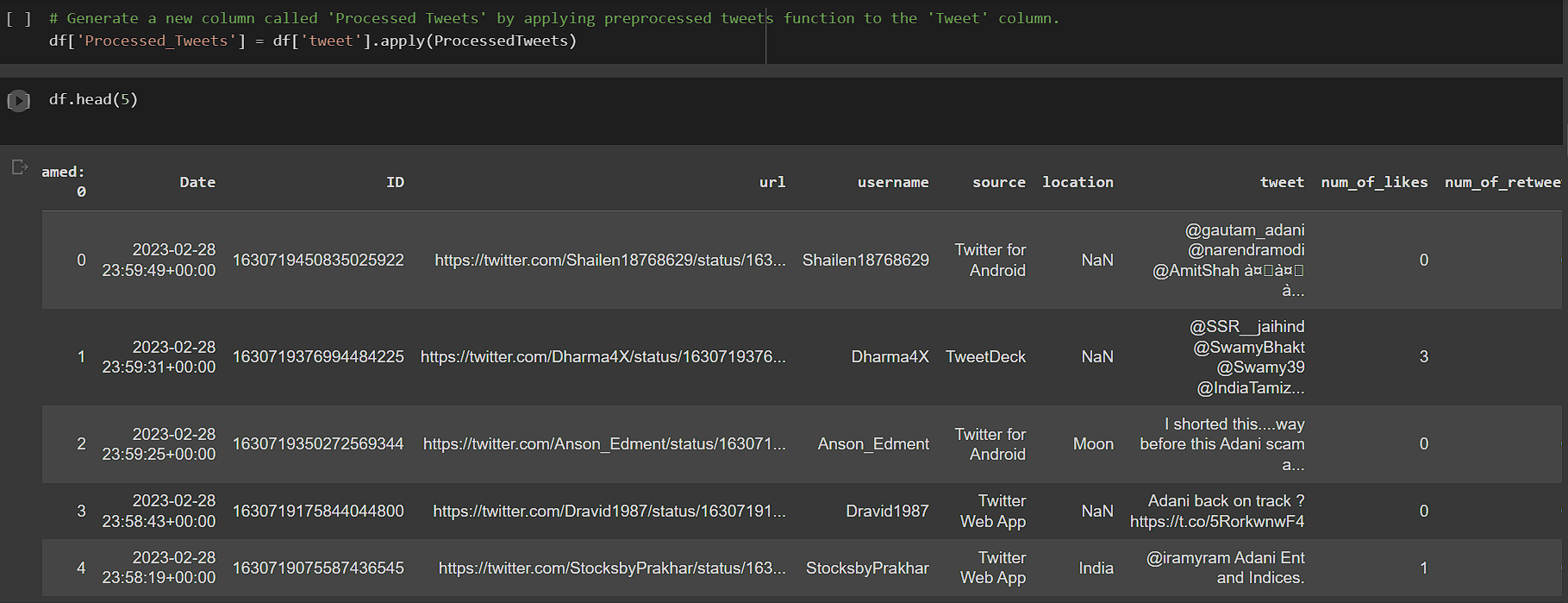
For Ex: All the words working, works, worked falls under the bracket of “word.”

The act of stemming involves stripping a word down to its fundamental or root form. This is helpful when grouping words with similar meanings but differing inflections in text analysis tasks. You can use the nltk (Natural Language Toolkit) library in Python to do stemming.

We first import the PorterStemmer class for stemming along with the nltk library. Then, we specify a list of words to stem by example. We build a PorterStemmer object, a well-liked stemming method, however nltk also includes additional stemming algorithms. Each word's stem is obtained by applying the stem() method of the stemmer object to it as we cycle through the words. The stemmed words are then printed as a list of strings.



PROCESSED TWEETS IN THE DATASET:



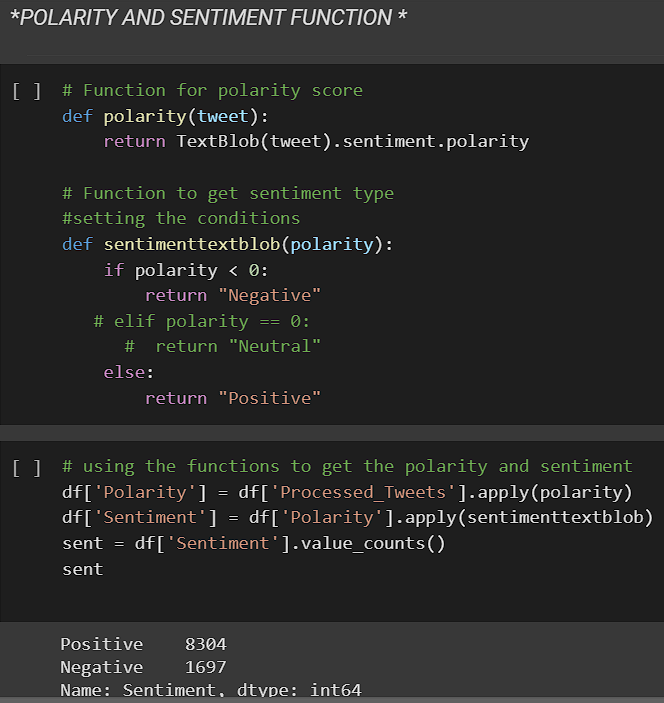


Fig. I Image of processed tweets



## Data Visualization

Any data science effort, including sentiment analysis, must include data visualisation. You can learn more about the data and more clearly convey your conclusions to others by visualising it. Here are some illustrations of data visualisations that can be helpful in a project including sentiment analysis:

1. WORDCLOUD: The size of each word in a word cloud indicates how frequently it appears in the text, and it is a visual representation of text data. The most frequently occurring words or topics in the text data can be found using word clouds.

2. HISTOGRAMS: A histogram is a graphic depiction of a numerical variable's distribution. A histogram can be used in the context of sentiment analysis to display the distribution of sentiment ratings for set of text data.

3. SCATTER PLOTS: are graphs in which data points are represented by dots. A scatter plot can be used to show the relationship between two variables in sentiment analysis, such as sentiment score and text length.

4. HEAT MAPS: A heat map is a type of graph in which the values of the data are represented by colours. A heat map can be used to show the prevalence of specific words or themes throughout a collection of text data in the context of sentiment analysis.

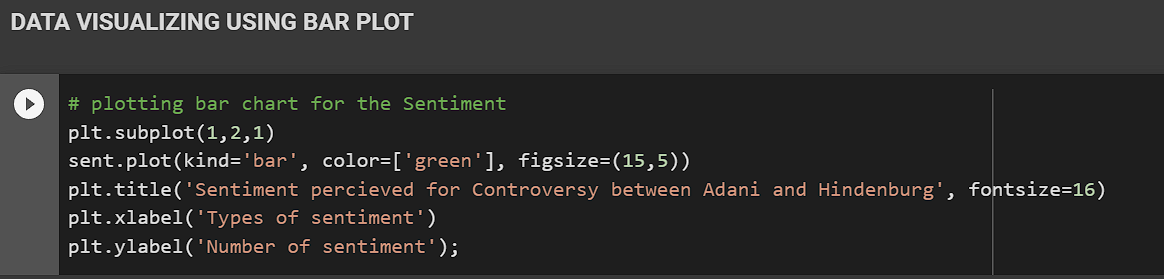
5. LINE CHARTS: Line charts are a type of graph that depicts changes in data over time. A line chart can be used to show how sentiment scores fluctuate in the context of sentiment analysis.

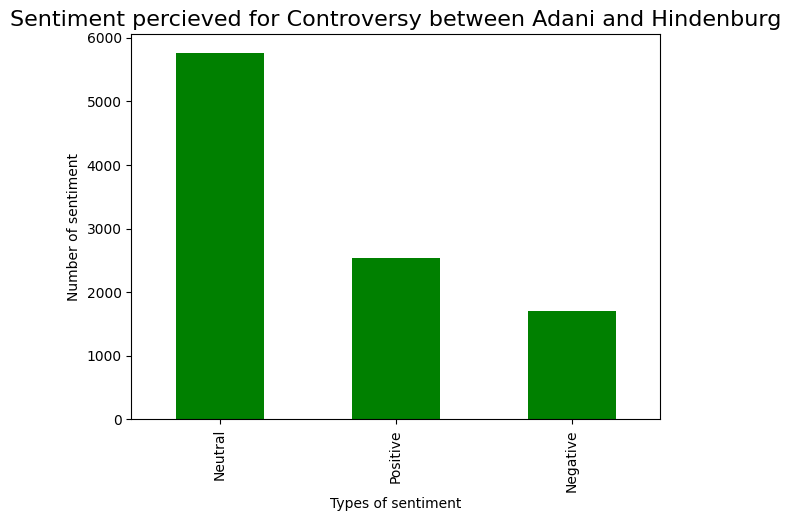
6. BAR CHARTS: In a bar chart, the length of the bars represents the frequency or value of a variable, respectively. A bar chart can be used in the context of sentiment analysis to compare the sentiment scores of several categories of text data, such as various brands or items.

Data visualization is used to graphically explain the data using charts, plots, Word-cloud for a better understanding of the data.

Word-cloud is used here which is a visualization tool wherein the most frequent word comes up in a large size and less frequent comes up in smaller size.

We can also mask any image into the frequent words of a Wordcloud and customize it accordingly.





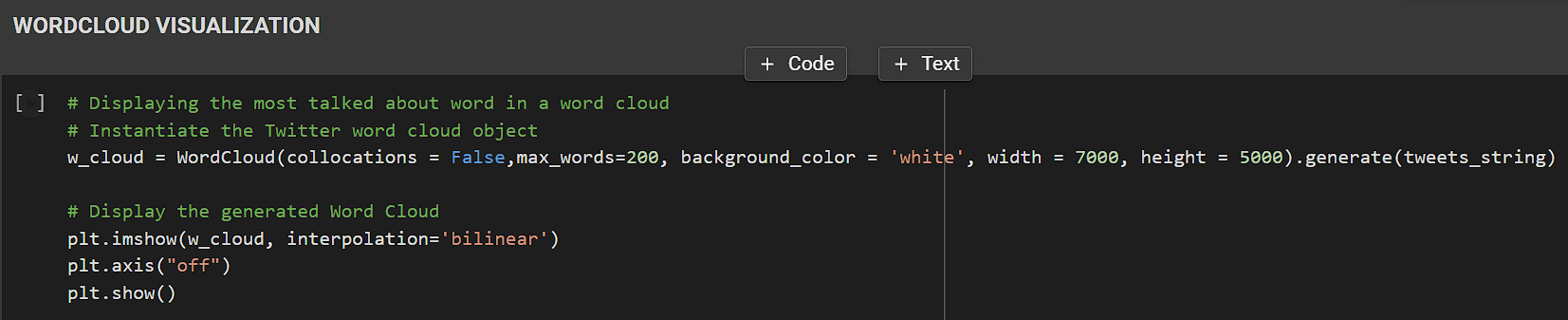
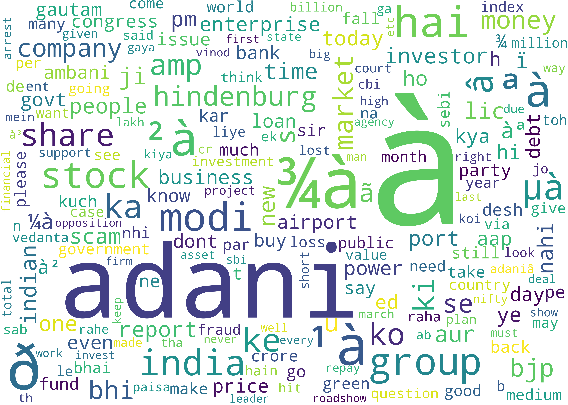
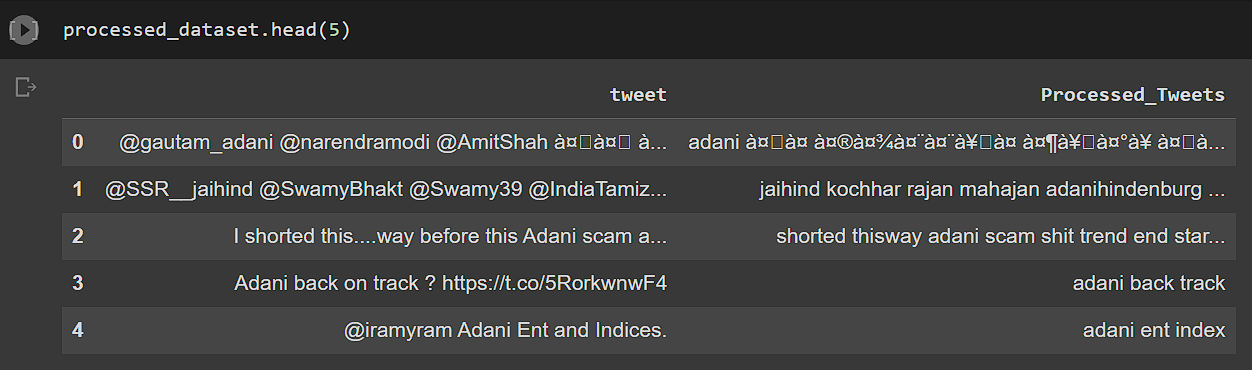


FIG. II WORLD CLOUD

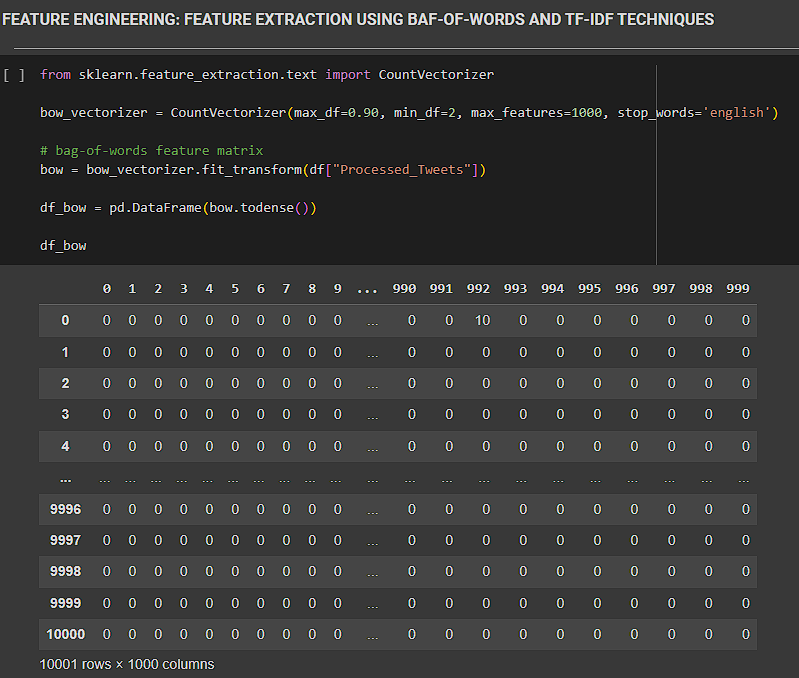


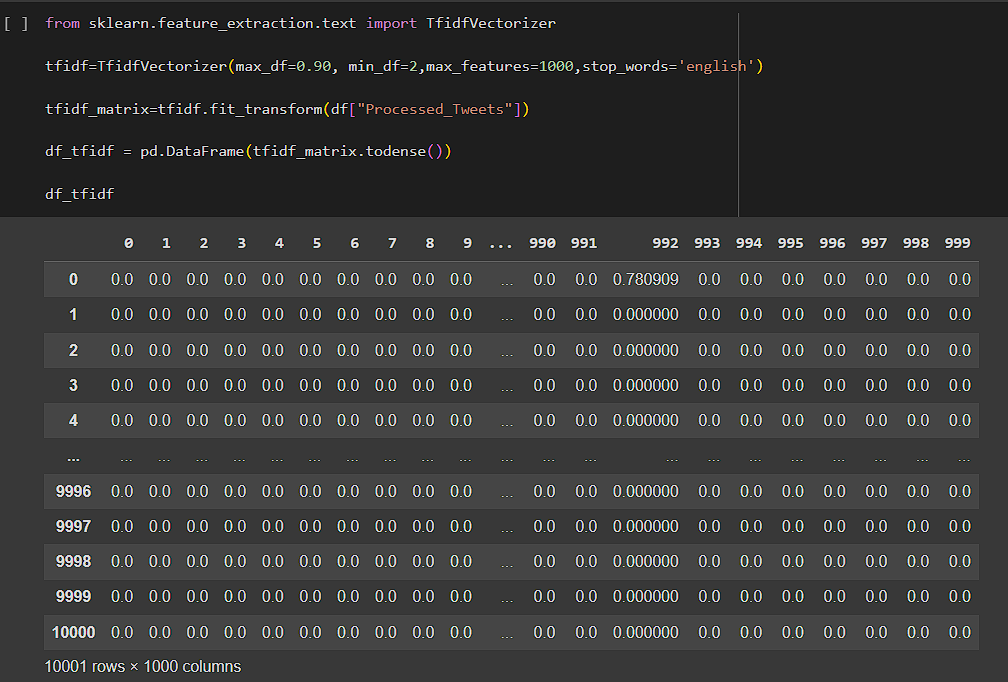






## Extracting features from cleaned tweets





* BAG-OF-WORDS FEATURES

Natural language processing (NLP) methods like the bag-of-words methodology frequently represent text as a collection of unorganized words or concepts without considering their order in the original text.

In this method, each word or phrase in a document or corpus is counted for frequency before a vector is created to represent the text or corpus.

Here is an example of how the bag-of-words technique works:

Let us say we have a group of three documents:

Document 1: “The quick brown fox”

Document 2: “Jumped over the lazy dog”

“The fox is quick and the dog is lazy,” says document 3.

To use the bag-of-words method, we first compile a vocabulary of every distinct word found in the group of documents. Our vocabulary in this situation would be: [“The,”“quick,”“brown,”“fox,”“Jumped,”“over,”“lazy,”“dog,”“is,”“and”]

After that, each document is represented as a vector of word frequencies.

This vector shows that in Document 1, the term “the,”“quick,”“brown,” and “fox” all appear once each, whereas all other words appear zero times.

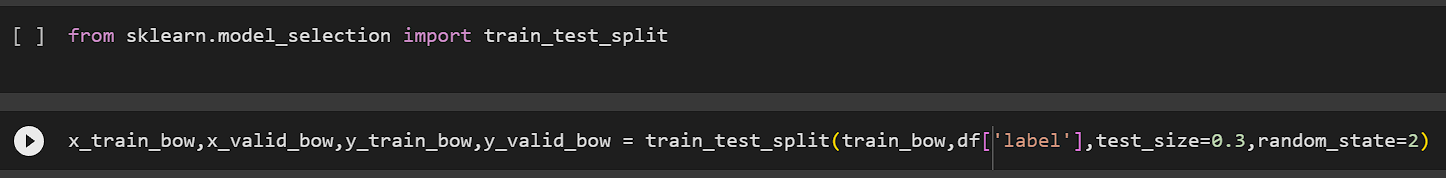
This vector shows that in Document 2, the phrases “Jumped” appears once, “over” appears once, “lazy” appears once, and “dog” appears once, while all other words appear zero times [TABLE III].

By representing each document as a word frequency vector, we can use these vectors for various NLP tasks such as text classification, pattern recognition, and data retrieval.

TABLE III VECTORIZATION OF VOCABULARY

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | The | Quick | Brown | Fox | Jumped | Over | Lazy | Dog | Is | And |
| D1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| D2 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| D3 | 2 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 2 | 1 |

# 



# TF-IDF FEATURES

The acronym TF-IDF stands for Term Frequency-Inverse Document Frequency.

It is a statistical technique used to assess a term’s significance inside a text or corpus (a collection of texts). The technique is founded on the notion that a term that regularly appears in a documentis essential, but the term may not be important if it frequently appears in numerous documents.

Here’s an example to illustrate the concept of TF-IDF:

Suppose we have a corpus of three documents:

EXAMPLE 1: “The cat in the house.”

EXAMPLE 2: “The cat saw the rat.”

EXAMPLE 3: “The dog ate the cat’s hat.”

We determine the importance of “cat” word in these documents using TF-IDF.

Step 1: Term Frequency (TF) for each “cat” word appearance

In ex 1, “cat” appears once.

In ex 2, “cat” appears once.

In ex 3, “cat” appears twice.

Step 2: Inverse document frequency (IDF) for “cat” appearance.

No. of example containing word “cat” =3

IDF formula: IDF = log(N/n), where N is the total number of documents/examples in the corpus and n is the number of documents/examples containing the term. In this case, IDF(“cat”) = log (3/3) = 0.

Step 3: TF-IDF score for “cat” in each document.

For ex 1, the TF-IDF score for “cat” is TF (“cat”, Document 1) \* IDF(“cat”) = 1 \* 0 = 0.

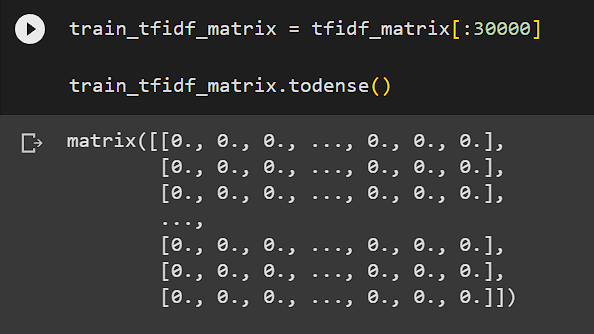
For ex 2, the TF-IDF score for “cat” is TF (“cat”, Document 2) \* IDF(“cat”) = 1 \* 0 = 0.

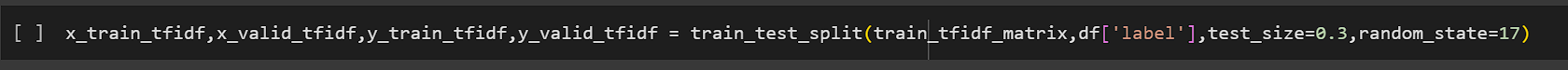
For ex 3, the TF-IDF score for “cat” is TF (“cat”, Document 3) \* IDF(“cat”) = 2 \* 0 = 0.

The results clearly show that “cat” is not a significant word in the corpus because it occurs quite frequently in all three documents.

Also,it’s IDF value is 0, which means it is not unique to any particular document.

In conclusion, TF-IDF is an effective method for locating the key phrases in a document or corpus. It can assist increase the accuracy of text-based applications like search engines and recommendation systems by assigning more weight to phrases that are specific to a document and less weight to terms that are widespread throughout the corpus.





## MACHINE LEARNING MODEL

Machine learning models are created by training algorithms using labeled or unlabeled data. Therefore, machine learning algorithms can be trained and produced in three ways: a) Supervised learning. b) Unsupervised learning. c) semi-supervised learning method. We used supervised learning to treat the algorithm.

Supervised Machine Learning is the category in which we are going to solve the underlying problem. As we take all labeled data for analysis of the tweets.

With supervised learning, you have input variables (x) and output variables (Y) and you use an algorithm to learn the mapping function.

Y=f(X)

Ideally, you want your mapping function to be approximated sufficiently well so that you can correctly predict the output variables (Y) when you have new input data (x).

To predict results on the test data, we generally use different models to see which one fits the dataset best.

1. LOGISTIC REGRESSION:

The only first model we are going to use in this analysis is Logistic Regression. This method is for a statical approach where we learn for binary classification problems. The goal is to predict whether the input belongs to one of two classes. The result is usually defined as 0 or 1.

The sigmoid function, commonly known as the logistic function, has the following form:

f(x) = 1 / (1 + e^(-x))

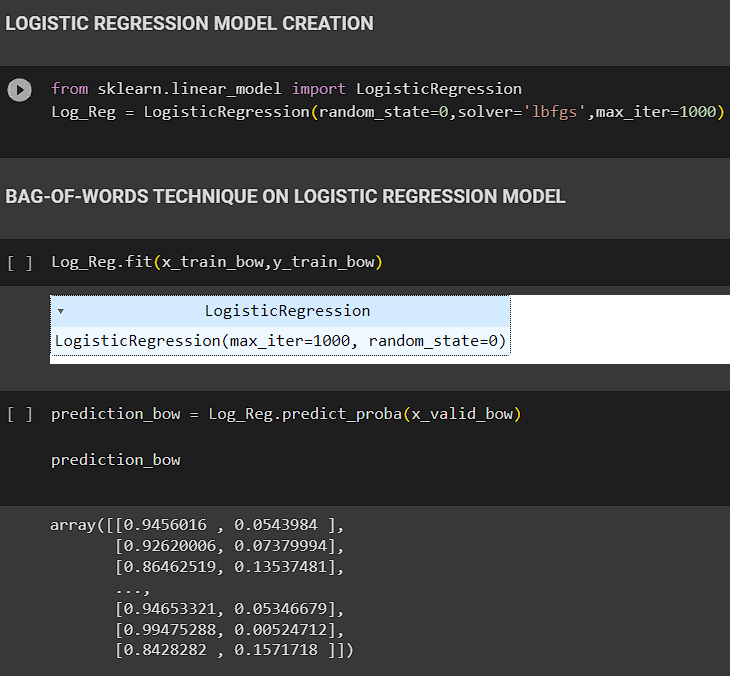
where f(x) is the anticipated probability of the event occurring and x is the linear combination of predictor variables.

By determining the values of the coefficients that maximize the likelihood of witnessing the data, maximum likelihood estimation is used to estimate the coefficients in the linear equation.

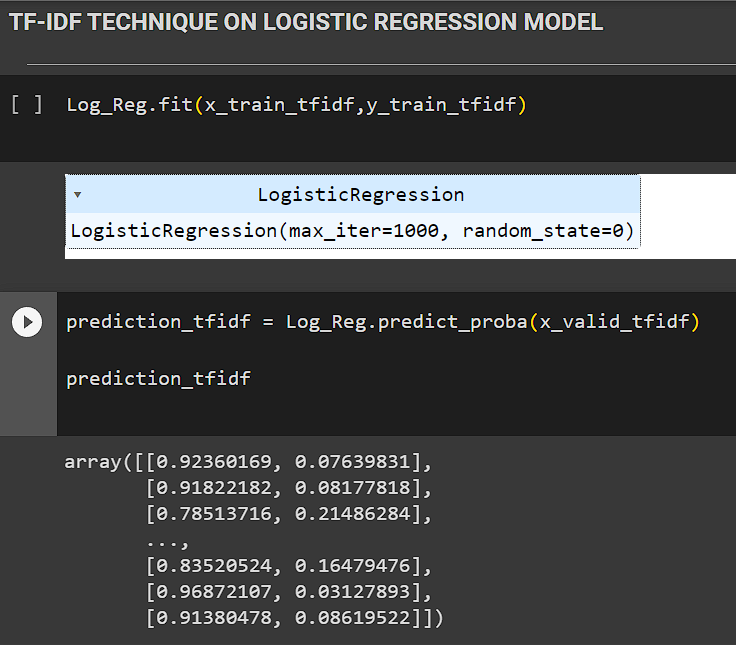
The logistic regression likelihood function is expressed as

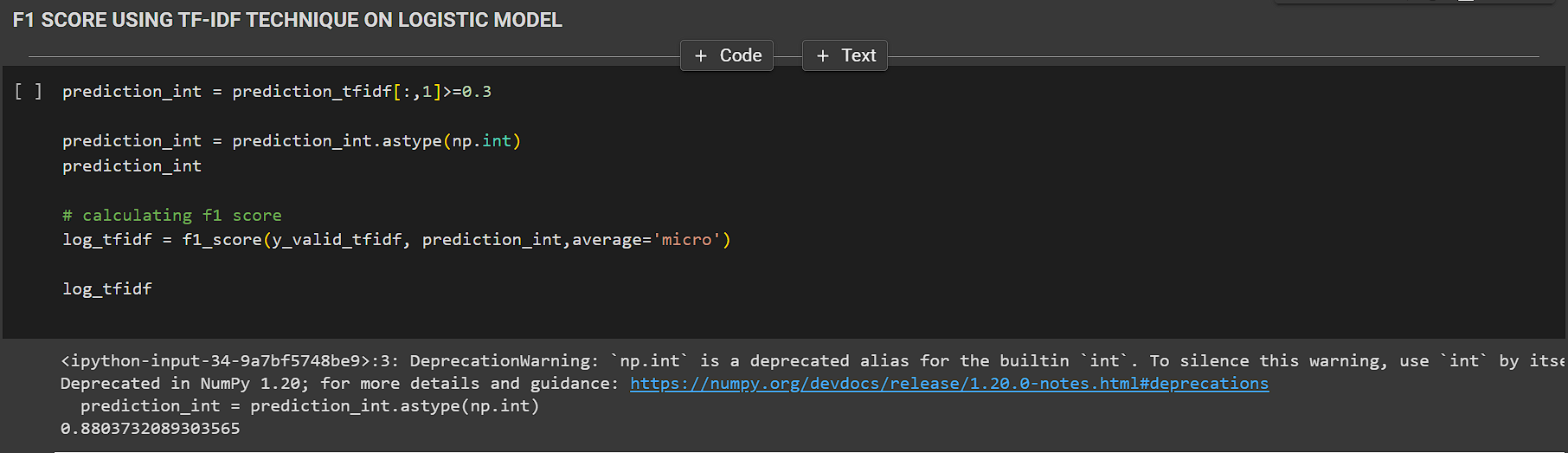
L = Σ (f(xi)yi \* (1-f(xi)) (1-yi)).

Where prod\_i stands for the sum of all observations, yi is the binary response variable (0 or 1) for the ith observation, and xi is the linear combination of predictor variables for the ith observation.









1. DECISION TREES:

The second model we used is decision trees.

Decision tree might have a node that tests whether the text contains the word “good”, and if so, branches to a positive sentiment node. If the text contains the word “bad,” another node might branch to a negative sentiment node.

As a splitting criterion, the property with the biggest information gain or the lowest impurity measure is chosen. Entropy, Gini index, and classification error are the three impurity measurements that are most frequently used.

Entropy is an indicator of how disorderly or uncertain a collection of samples is, and it is defined as:

Σ (p\_i \* log2(p\_i)) = - H(S)

S stands for the sample set, pi represents the percentage of samples that correspond to class i, and log2 stands for the binary logarithm.

The difference between the set’s entropy before and after the attribute A split is used to define the information gain of an attribute A with regard to a set of samples S:

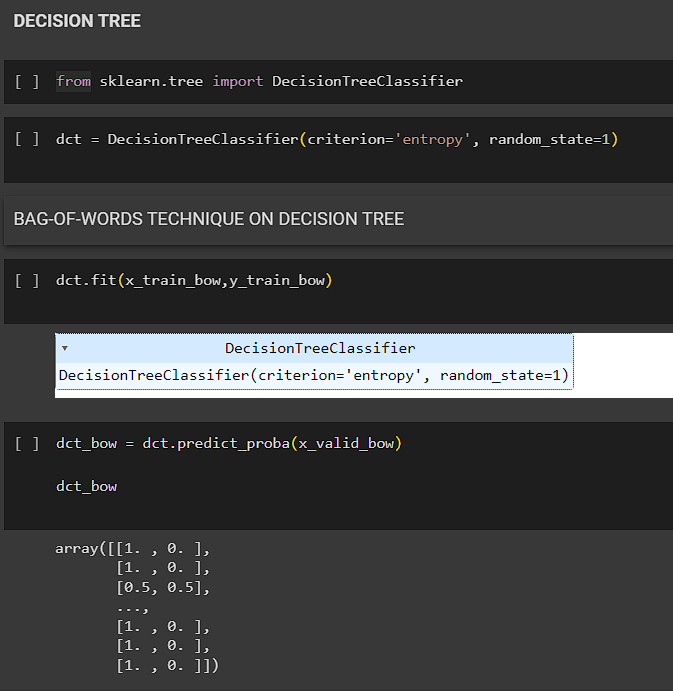
H(S) = Σ (|S\_v|/|S| \* H(S\_v)) –IG(A, S)

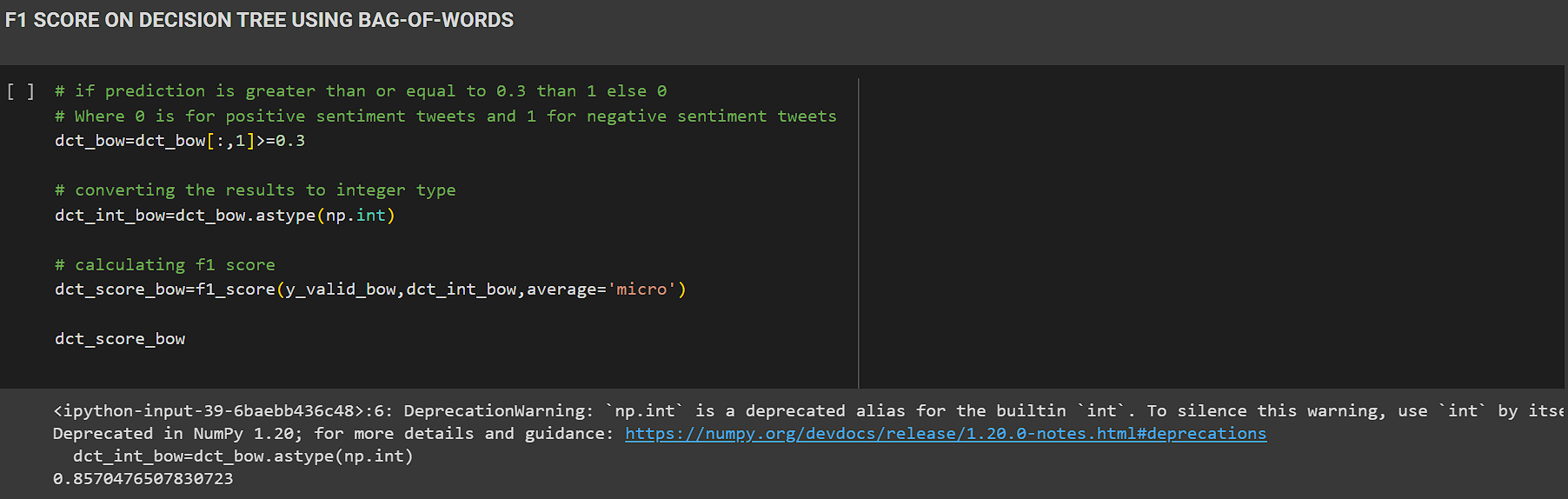
where |S| represents the number of samples in S, S\_v represents the subset of samples where attribute A = v, and v is the value of attribute A.

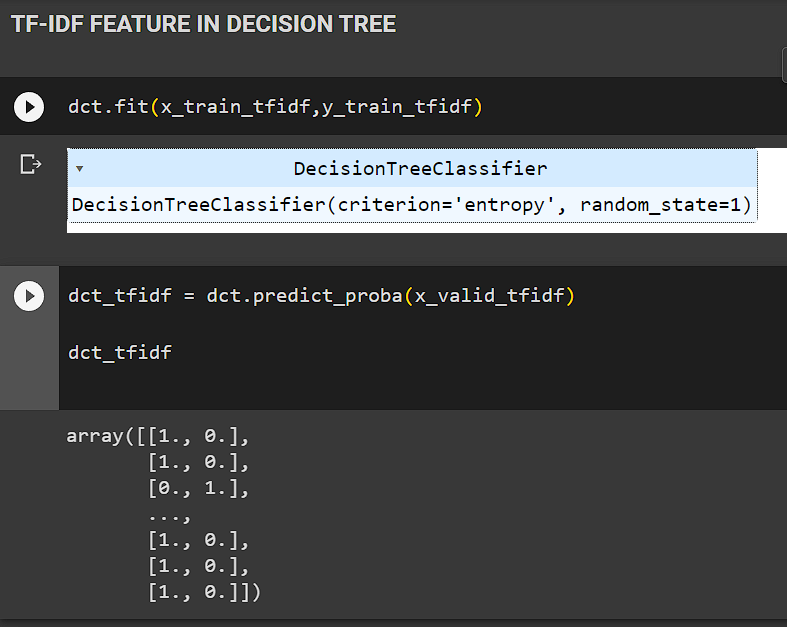
The Gini index is an indicator of sample impurity and is defined as:

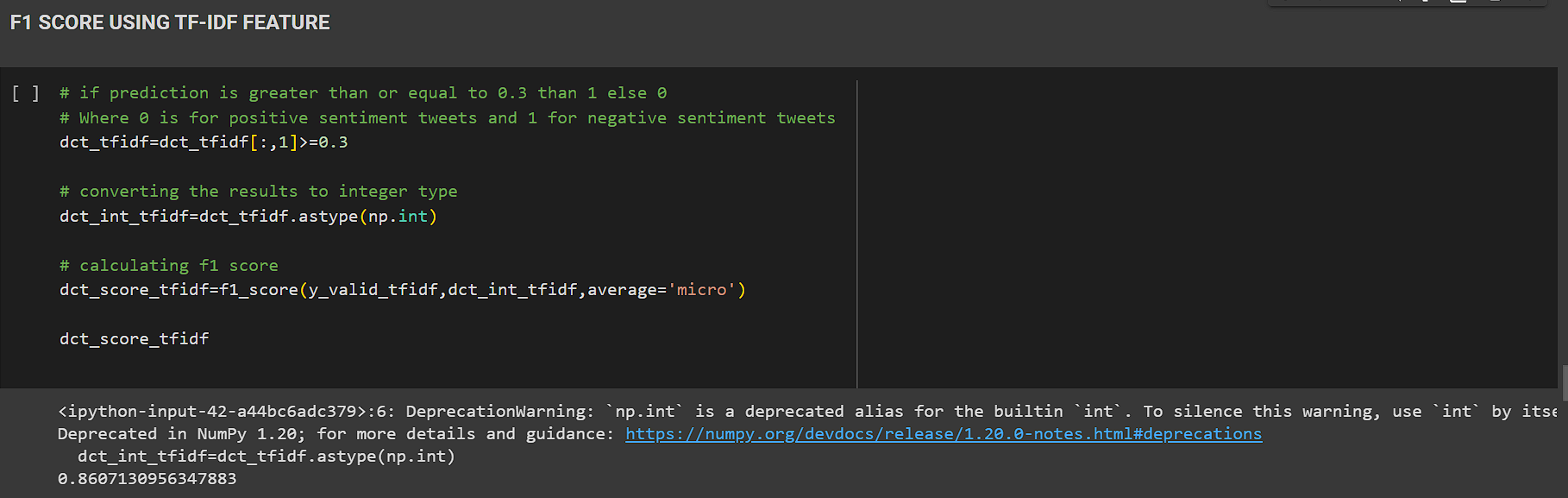
Sum\_i (p\_i2) – 1 equals Gini(S).

where pi represents the percentage of samples that correspond to class i.









1. NAÏVE BAYES CLASSIFIER:

Naive Bayes can be learned quickly and is computationally effective on big datasets. Due to this, many applications, such as spam filtering and sentiment analysis, favor it.

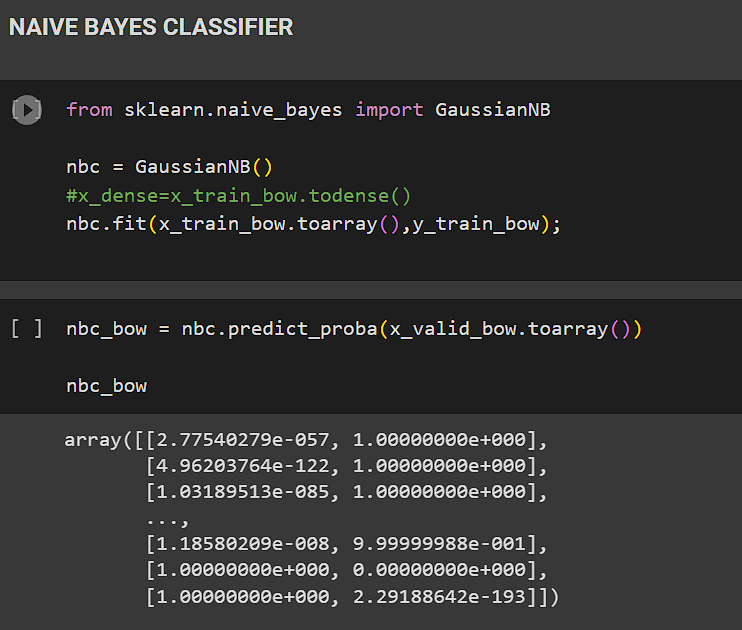
It states that, normalized by the probability of E, the probability of a hypothesis H given some evidence of E is proportional to the sum of the prior probability of H and the likelihood of E given H:

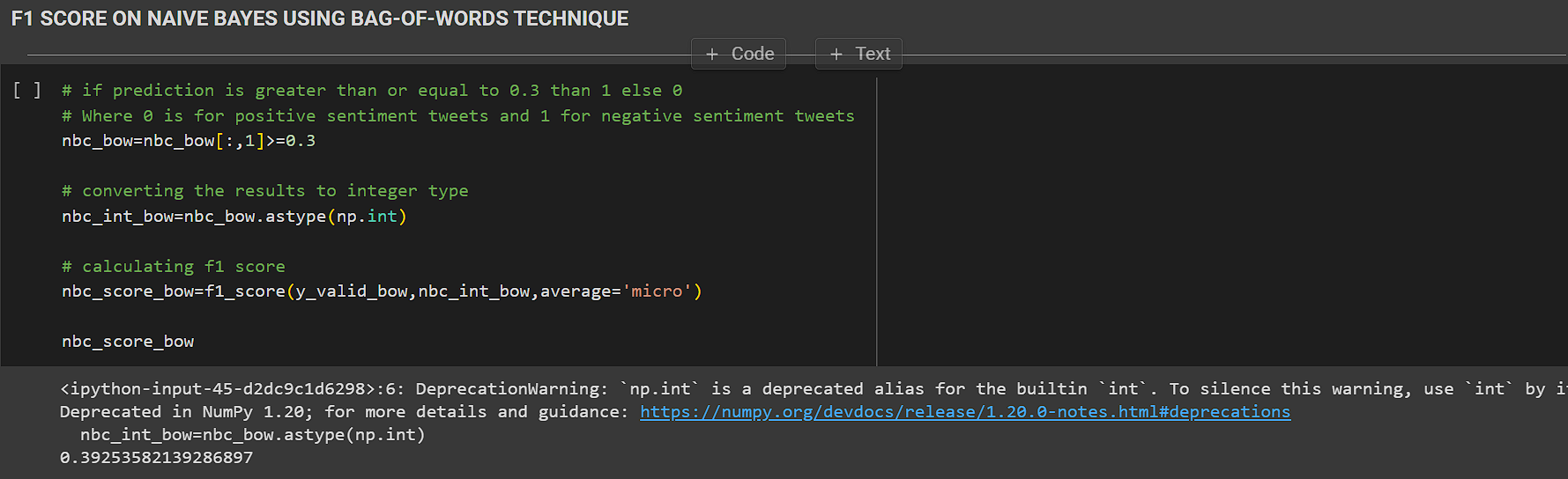
P (H | E) = P(E | H) \* P(H)

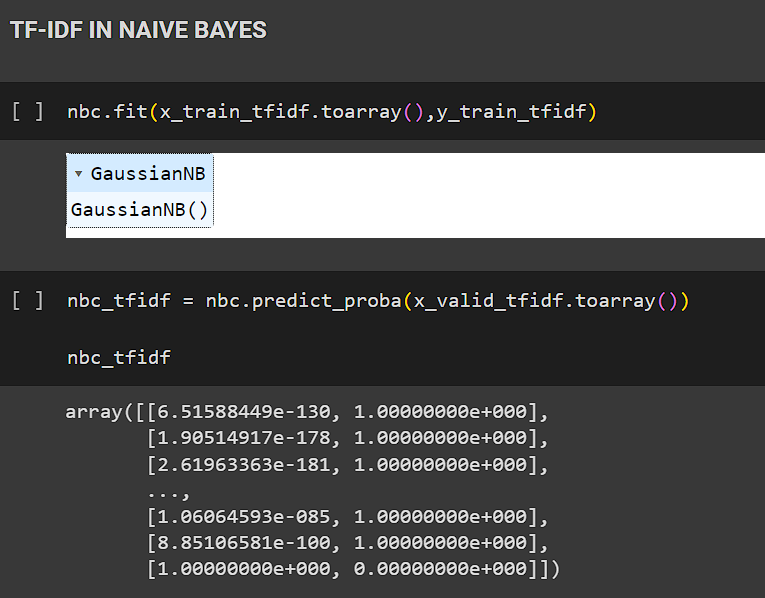
P(E)

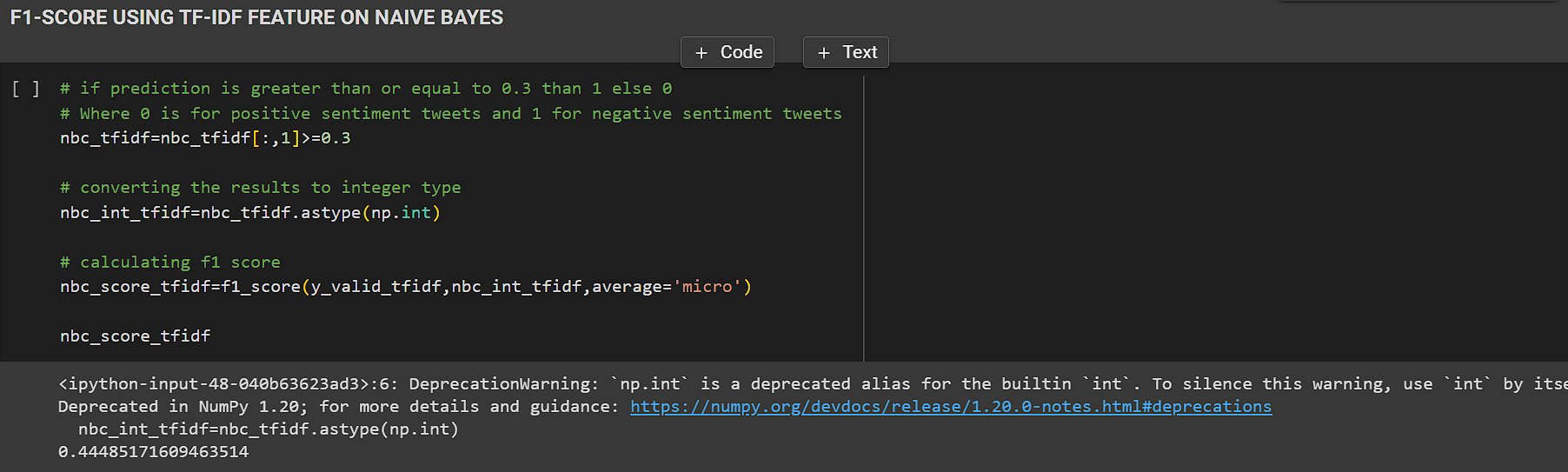
By assuming that the features are conditionally independent given the class label, Naive Bayes may separately calculate the probability of each feature given the class label:

P (X\_1, X\_2, …, X\_n | C) = P(X\_1 | C) \* P(X\_2 | C) \* … \* P(X\_n | C)







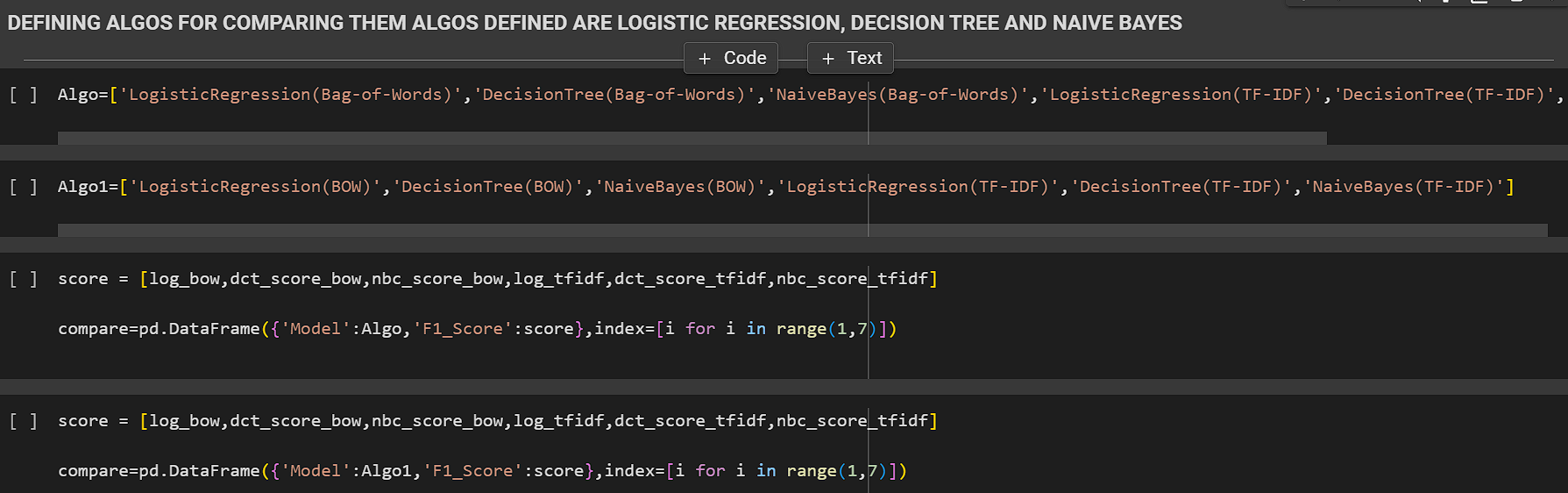


# EVALUATION AND RESULTS

In this paper we have used dataset which consists of 10,000 tweets generated on our specific query. We later go through the data science lifecycle of data cleaning, data pre-processing, EDA, feature engineering techniques used like Bag-of-Words and TF-IDF, then building three supervised machine learning models – Logistic Regression, Naïve Bayes and Decision tree for training and testing our data.

We have first trained our models on bag of words technique and later used TF-IDF technique.

For logistic regression we obtained the F1 score as0.877 in Bag of words technique and F1 score as 0.8803 in TF-IDF technique. Similarly, f1 for naïve bayes were 0.392 and 0.444, f1 score for decision tree were 0.857 and 0.860 respectively [TABLE IV].



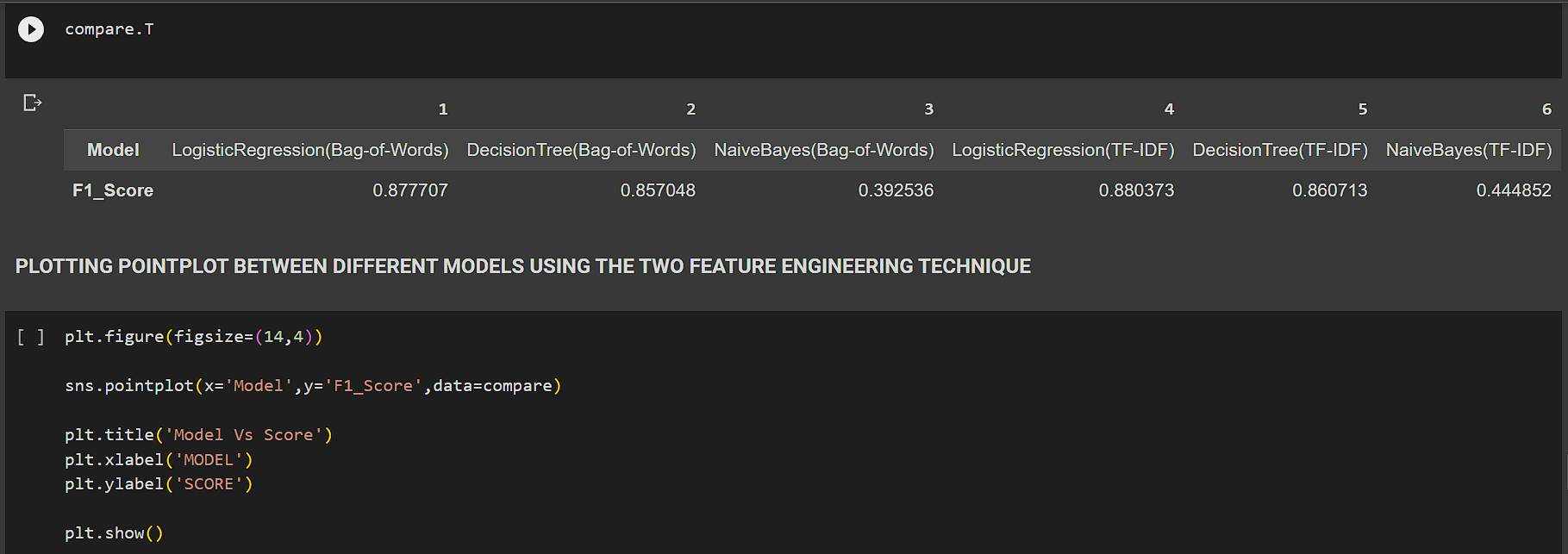


TABLE IV COMPARATIVE ANALYSIS OF F1 SCORES

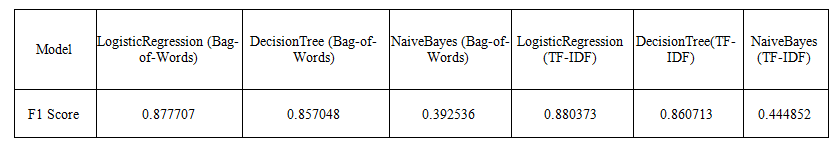
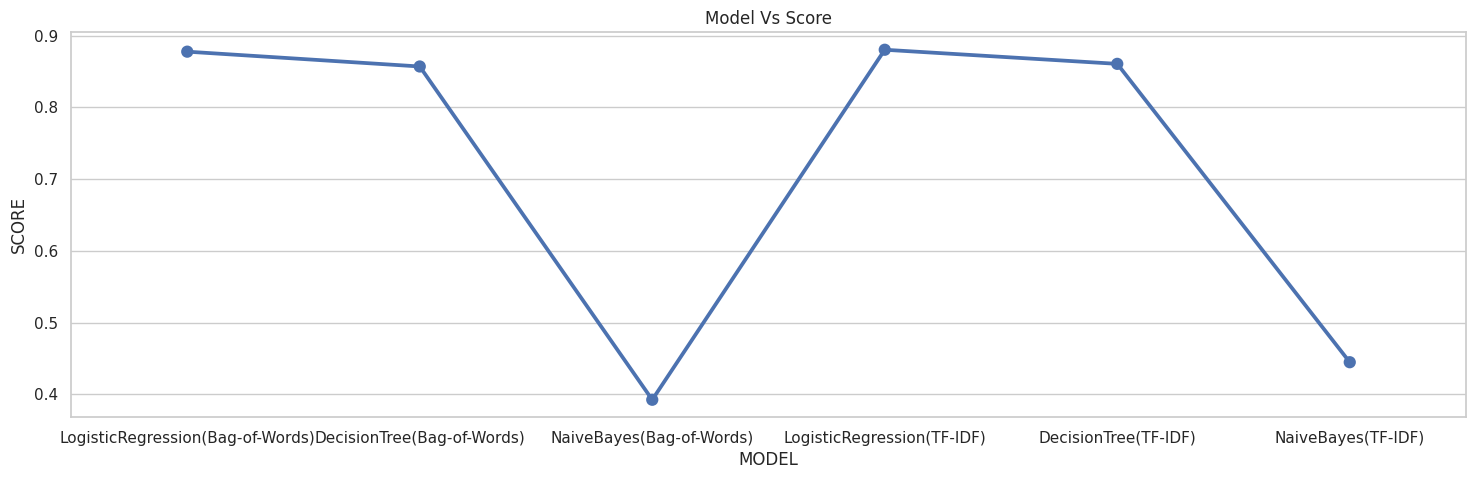


Fig. III MODEL vs SCORE

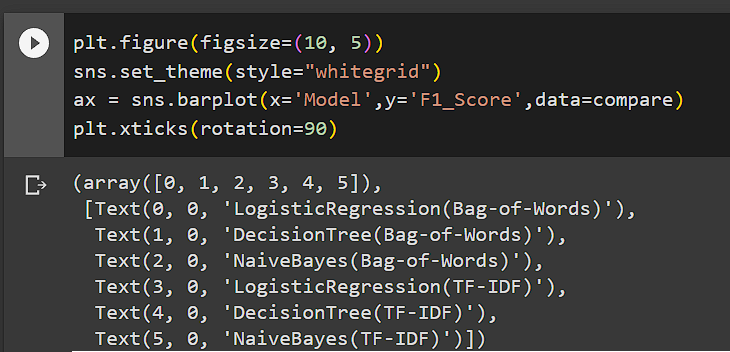


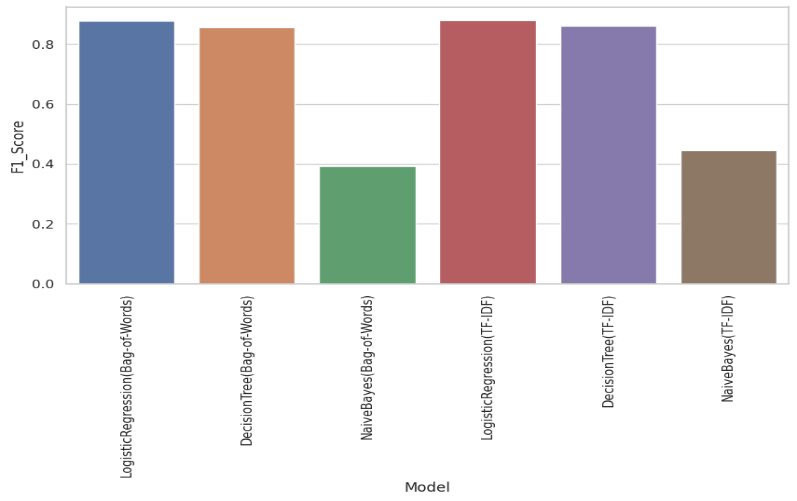
Herein we used F1 score as the resultant metric. Depending on the problem you are trying to solve, you could often either assign a larger premium to optimizing precision or recall. However, there is a more straightforward statistic that generally accounts for both recall and precision, so you can attempt to increase this number to improve your model. The harmonic mean of Precision and Recall is known as the F1-score, which is the measure in question.

F1 Score = 2 \* Precision \* Recall

Precision + Recall

Fig. IV F1 SCORE WITH TF-IDF





From our experiments with various models and feature engineering techniques we came to the conclusion that Logistic Regression gives the best result or best F1 Score with TF-IDF technique followed by Bag-of-Words technique followed by decision tree model and the worst result was obtained through Naïve Bayes model.

# **CONCLUSIONAND FUTURE SCOPE**

We learned from this study about the various groups to which our sentiment ratings belong, both polarity- and subjectivity-wise. We can aggregate tweets that are positive, negative, and neutral by creating a cluster of the results from both tools' scores since pre-defined dictionaries or sentiment technologies can't accurately score every word in the context of a phrase. Some tweets only contain news or other people's opinions on a particular expression, but once we identify the subjectivity (opinion, feeling, or emotion of a specific person expressing it) of a tweet, we can group it using sentiment scores to determine the ratio of the true positive, negative and neutral "opinions" and not just sentiments of a sentence. Future development and growth in the field of sentiment analysis are highly anticipated. More complex sentiment analysis models are required to analyse sentiment accurately in various languages, situations, and data sources due to the growing demand for sentiment analysis across numerous industries.