

# INDIAN STATISTICAL INSTITUTE

## POST-GRADUATE DIPLOMA IN STATISTICAL METHODS AND ANALYTICS



Final Year Project

---

NLP: SENTIMENT ANALYSIS ON YOUTUBE COMMENTS FOR SAMSUNG  
BRAND IMAGE IMPROVEMENT

---

**Authors:**

**SRIMANTA SINGHA**  
(Roll:DSTC-22/23-017)

**MOHAMMAD ABDUL  
WAHED**  
(Roll:DSTC-22/23-010)

**Supervisors:**

**DR.SUDHEESH KUMAR  
KATTUMANNIL**  
Indian Statistical Institute  
Chennai

May 30, 2023

---

## CERTIFICATE

---

This is to certify that the document titled ”**SENTIMENT ANALYSIS ON YOUTUBE COMMENTS FOR SAMSUNG BRAND IMAGE IMPROVEMENT**” , submitted by **Srimanta Singha & Mohammad Abdul Wahed** to Indian Statistical Institute is a record of the the project they did under my supervision and was carried out by them for the partial fulfillment of the requirements for the Post-Graduate Diploma in Statistical Methods and Analytics programme of the Indian Statistical Institute.

(Supervisor’s Signature)

**DR.SUDHEESH  
KUMAR**

**KATTUMANNIL**

Indian Statistical Institute  
Chennai

---

## Acknowledgements

---

We feel lucky to be guided by Dr.Sudheesh Kumar Kattumannil and thank him for his encouragement, advice, and care. We also appreciate his way of questioning, conceptualizing, and thinking, which made us focus onto more detailed understanding, while never forgetting to look into the data handling part of the analysis. We would like to express our heartfelt gratitude and respect to Dr. G. Ravindran, Head of the ISI Chennai Centre for his guidance from time to time. We thank all other faculty members for their encouragement and assistance. We shall remain ever indebted to our parents for always motivating and supporting us in every aspect of our life and career.

Date: 29/05/2023

Srimanta Singha & Mohammad Abdul Wahed.

## **Abstract**

Over time, textual information has increased exponentially, resulting to the potential research within the field of machine learning (ML) and natural language processing (NLP). Social media websites are some of the world's most popular websites and allow all users to have a voice and express opinions and emotions. Nowadays, YouTube is one of the most popular social media platform where people can share their opinions, emotions to a particular issue by commenting on corresponding videos. Using sentiment analysis, these users' opinions and emotions can be extracted and quantified. We all know that Samsung group, a South Korean company that is one of the world's largest producers of electronic devices, are launching the new smartphone "Galaxy S23 Ultra" and they already gave many official trailers on YouTube. In this project, we perform sentiment analysis on the YouTube comments about the smartphone "Samsung Galaxy S23 Ultra" using NLP and machine learning techniques/algorithms to improve brand image of this smartphone.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Objectives . . . . .	2
<b>2</b>	<b>Background</b>	<b>3</b>
2.1	YouTube . . . . .	3
2.2	NLP: Natural Language Processing . . . . .	5
2.2.1	Use of ML in NLP . . . . .	6
2.2.2	Application of NLP . . . . .	6
2.3	Sentiment Analysis . . . . .	7
<b>3</b>	<b>Methodology</b>	<b>8</b>
3.1	Data Gathering . . . . .	8
3.2	Data Pre-processing . . . . .	8
3.2.1	Features selection . . . . .	9
3.2.2	Data labelling . . . . .	9
3.2.3	Data Cleaning & Data Transformation . . . . .	9
3.3	Data Splitting . . . . .	10
3.4	Creating ML Model . . . . .	10
3.5	Conclusion . . . . .	11
3.6	framework of python and deploy it on Heroku platform . . . . .	11
<b>4</b>	<b>ML Algorithms used on Sentiment Analysis:</b>	<b>12</b>
4.1	Naïve Bayes classifier: . . . . .	12
4.2	Support Vector Machine(SVM): . . . . .	15
4.3	Random Forest: . . . . .	17
4.3.1	Assumptions for Random Forest: . . . . .	18

4.3.2	Why use Random Forest? . . . . .	19
4.3.3	Random Forest Algorithm: . . . . .	19
4.4	Model Performance Checking: . . . . .	21
<b>5</b>	<b>Performing Sentiment Analysis of YouTube comments on Samsung Galaxy S23 Ultra's official trailer:</b>	<b>24</b>
5.1	Data collect from YouTube: . . . . .	24
5.2	Data Pre-processing: . . . . .	26
5.2.1	Features selection: . . . . .	26
5.2.2	Deleting Non-English comments: . . . . .	26
5.2.3	Data Labelling: . . . . .	27
5.2.4	Data cleaning & Transformation: . . . . .	28
5.2.5	Label Encoding: . . . . .	32
5.3	Balancing the dataset: . . . . .	33
5.4	Corpus & Count vectorization: . . . . .	36
5.5	Splitting & Machine learning models building: . .	38
5.5.1	Gaussian Naive Bayes Classification: . . . .	38
5.5.2	Random Forest Classification: . . . . .	39
5.5.3	Support Vector Machine Classification: . .	40
5.5.4	Accuracy Table: . . . . .	41
5.5.5	Prediction Function: . . . . .	41
<b>6</b>	<b>Model Deployment</b>	<b>43</b>
<b>7</b>	<b>Conclusion</b>	<b>49</b>
<b>8</b>	<b>Biblography</b>	<b>51</b>

# 1 Introduction

In this work, we will collect the comments of the public on the YouTube official trailer of "Samsung Galaxy S23 Ultra" and measure the attitude of the user towards the aspects of the phone which they describe in a text.

Sentiment analysis is useful for quickly gaining the whole idea by using large number of text data and it will be helpful to understand the user's opinion. Sentiment analysis is additionally referred as opinion mining that means to find out or identify the positive,negative, neutral opinions, views, attitudes, impressions, emotions and feelings indicated in the text.

As of 2023, YouTube is the second biggest social media in the world, with over 2.5 billion active users.Only Facebook (2.9 billion) has more active users than YouTube.There are 4.65 billion active social media users worldwide. This means that 54.34% of active social media users in the world access YouTube. That's why, With a large amount of user data available through YouTube, it is possible to gain insights into the users' opinions and emotions.The field of sentiment analysis is the computational study of people's emotions, opinions, and attitudes expressed in written text.With the use of statistical models such as machine learning(ML) and natural language processing(NLP), sentiment analysis can be performed to classify and quantify user emotions and opinions expressed on YouTube.

In this project, Sentiment analysis of YouTube comments about "Samsung Galaxy S23 Ultra" is carried out on the dataset containing 800 comments.There are some noise in our dataset. The noise is



to be cleaned from data using different data normalization rules in order to clean the comments from the dataset. To perform classification on this dataset we developed a system in which three different machine learning algorithms including Naïve-Bayes (NB), Support Vector Machine (SVM) and Random Forest (RF) are implemented. Accuracy of the classification algorithms has been evaluated using different evaluations measures e.g., F-Score and Accuracy score to evaluate the classification system's correctness. To improve our system's performance, we've used different features selection techniques like lemmatization, stop words and punctuation removal.

## **1.1 Objectives**

Sentiment analysis is the process of retrieving information about a consumer's perception of a product, service or brand. If you want to know exactly how people feel about your business, sentiment analysis is the key. Sentiment analysis is a machine learning technique that can analyze comments about your brand and your competition for opinion polarity (positive, negative, neutral). Here we performed sentiment analysis on YouTube comments about upcoming smartphone "Samsung Galaxy S23 Ultra" to know the sentiment of public on it. It helps the Samsung group in gauging market trends of their product in future.

## 2 Background

Before applying Sentiment analysis on our dataset(YouTube comments on "Samsung Galaxy S23 Ultra"),we will discuss about YouTube, NLP, Sentiment anlysis and some Machine learning algorithm which are going to use here.

### 2.1 YouTube

YouTube is a platform for distribution and consumption of videos and is one of the most popular websites and the social media platform with the second highest number of active users. Content creators consist of individuals, organizations, music artists among others. When a user/channel has published a video, other YouTube users can interact with the video. Users can give the video a like or a dislike, leave comments or reply to other comments. The maximum length of YouTube comments is 9999 characters, including spaces. YouTube trending is a collection of nationwide top lists updated every 15minutes, It aims to show among other things videos that are novel and "appealing to a wide range of viewers". Ranking indicators include view count, speed of views generation, age of the video, and the relative performance of a video compared to other videos from the same channel.

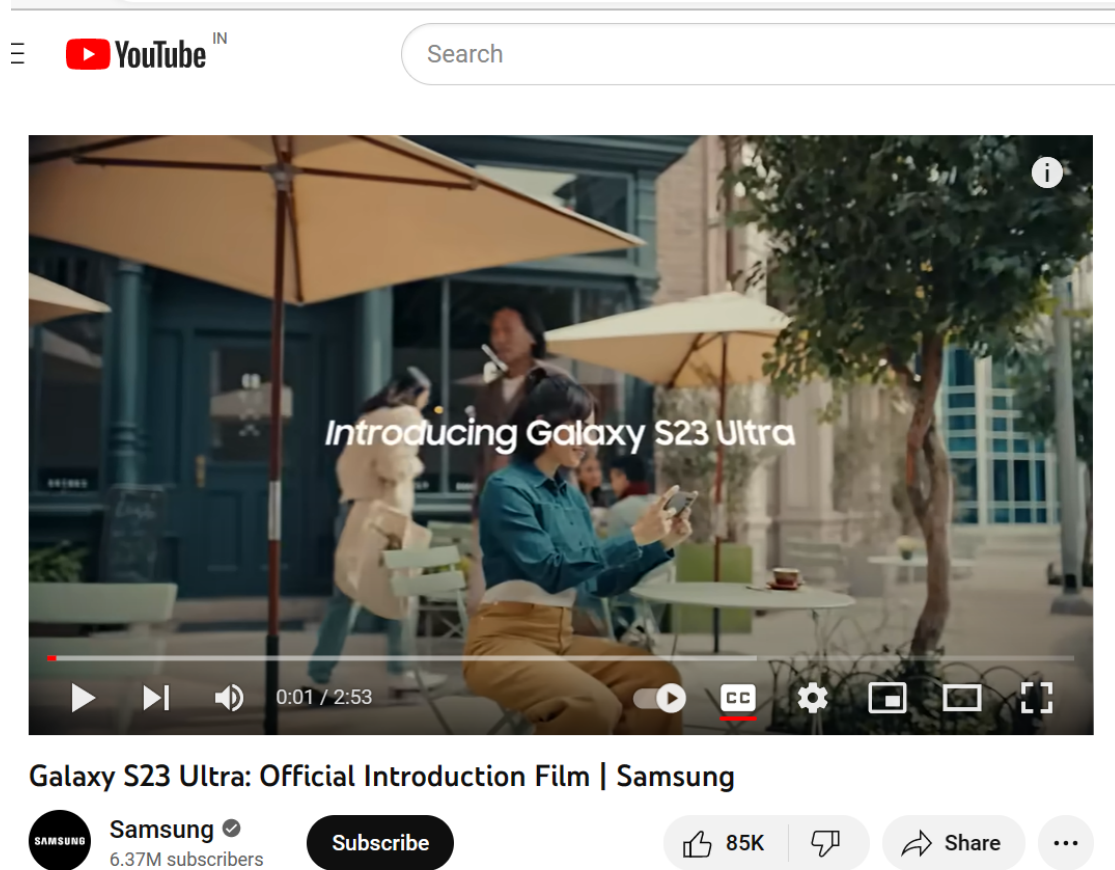


Figure 2.1: Samsung Galaxy S23 official trailer.

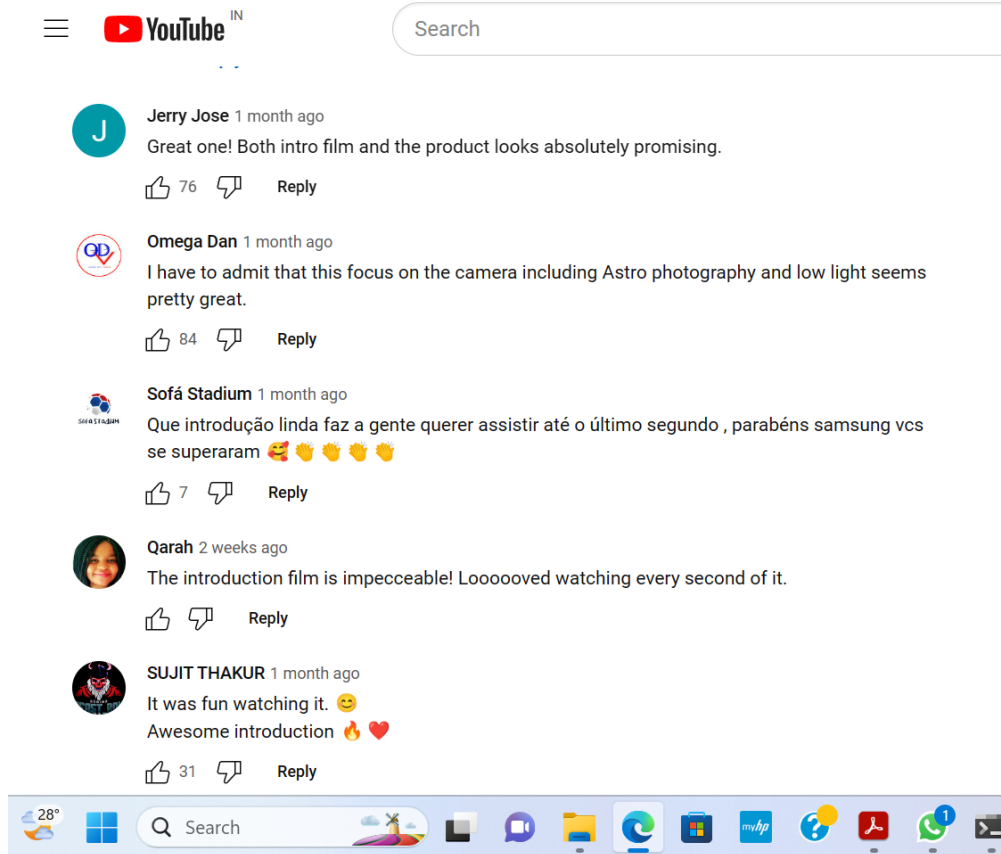


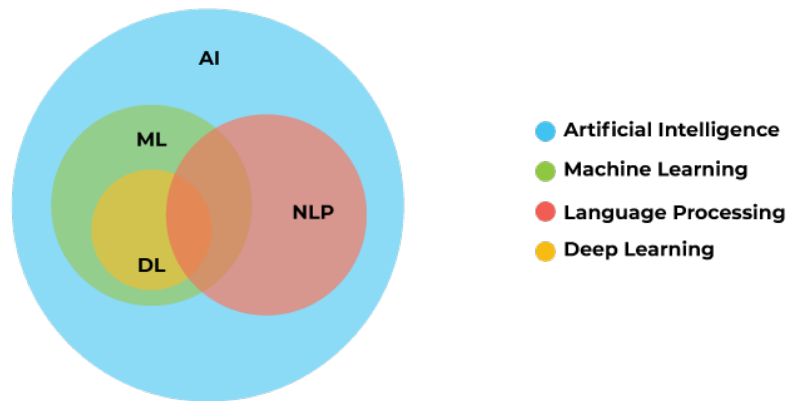
Figure 2.1: YouTube comments on Samsung Galaxy S23

## 2.2 NLP: Natural Language Processing

NLP stands for Natural Language Processing, which is a part of Computer science, Human language and Artificial intelligence. It is the technology that is used by machines to understand, analyse, manipulate and interpret human language. It helps developers to organize knowledge for performing tasks such as translation, automatic summarization, Named Entity Recognition (NER), speech recognition, relationship extraction, topic segmentation and sentiment analysis.

### 2.2.1 Use of ML in NLP

NLP and ML are both subsets of Artificial intelligence(AI).Machine learning is an application of AI that provides system the ability to automatically learn and improve from the experience. So ML can be used to improve NLP by automating process and delivering accurate responses. Here we can get some basic idea with the following diagram.

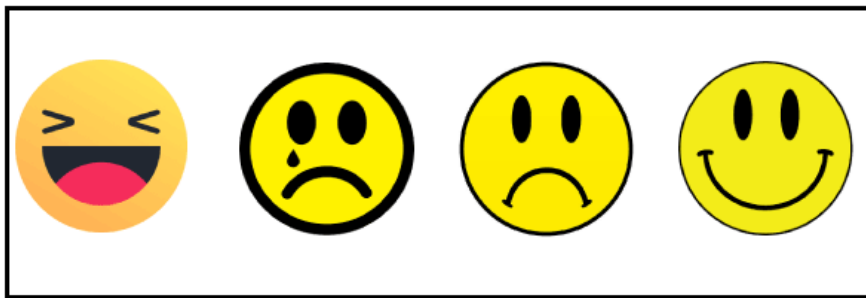


### 2.2.2 Application of NLP

NLP is a part of everyday life and it is essential to our lives at home at work. Without giving much thought, we send voice commands to our virtual home assistance, our smartphones and even our vehicles. Voice enabled applications such as ALEXA, SIRI and GOOGLE ASISTANT use NLP and Machine Learning(ML) to answer our questions. NLP is not only making our lives easier but revolutionizing the way we work, live and play. The most popular application of NLP are Question answering, Spam detection, Sentiment analysis, Spelling correction etc.

## 2.3 Sentiment Analysis

Sentiment Analysis is one of today's most popular application of NLP. Sentiment analysis is the process of classifying whether a block text is positive,negative or neutral.It is contextual mining of words which indicates the social sentiment of a brand and also helps the business to determine whether the product which they are manufacturing is going to make a demand in the market or not. It also identify the mood of the context (happy, sad, angry, etc.)



## 3 Methodology

In this section, we discussed about the material and methods which are used to do the Sentiment analyse the YouTube comments on upcoming smartphone "Samsung Galaxy S23 Ultra".

### 3.1 Data Gathering

To do the Sentiment analysis on YouTube comments, first we need to gather the YouTube comments on a certain issue, brand or product as dataset. One can use YouTube API key to scrape the YouTube comments on "Samsung Galaxy S23 Ultra" but in this project, we did not use the YouTube API key method to scrape the comments. Instead of YouTube API key, we used only Python library package for scraping the YouTube comments on "Samsung Galaxy S23 Ultra". After collecting the comments on "Samsung Galaxy S23 Ultra" from YouTube, we stored these comments as CSV format dataset.

### 3.2 Data Pre-processing

Some pre-processing of the data are necessary. Our chosen method could not handle all comments from the datasets without failing. Since the data files were read line by line, newlines () within the comments had to be removed. Certain emojis couldn't be properly encoded in our chosen file format (CSV UTF- 8) so those emoji characters had to be deleted. Here we have followed the procedure below.

### **3.2.1 Features selection**

Since we are going to use the Sentiment analysis on YouTube comments so only the column containing comments is the only important features of our work. In this section we reconstructed the dataset by dropping unnecessary columns from our original dataset and proceeded to next step with new dataset.

### **3.2.2 Data labelling**

Since our dataset containing the YouTube comments on "Samsung Galaxy S23 Ultra" is unlabeled dataset. To perform Sentiment analysis using Supervised machine learning algorithm, it is required to label the dataset as Positive, Negative and Neutral. We are labelling the new dataset with the help of polarity of the comments as well as python packages.

After labelling the dataset using python, there are some new columns appeared in our dataset due to polarity score of the comments, so we again dropped the unnecessary columns (polarity columns) from the dataset. Finally our dataset contained the columns of Comments and Sentiment.

### **3.2.3 Data Cleaning & Data Transformation**

Some cleaning of the data is necessary. Our chosen machine learning method and algorithm could not handle all comments from the datasets without failing. Some of things such as newlines (), Stop words, Punctuation, Multiple spaces, references and hastages and Special characters within the comments had to be removed. Certain emojis couldn't be properly encoded in our chosen file format (UTF-8) so those emoji characters had to be deleted.



After cleaning the dataset, now some transformation is needed to our dataset because directly text data cannot be given to machine learning algorithms, it should be converted into a suitable type. We used NLTK package named "WordNetLemmatizer" to break a word down to its root meaning to identify similarities. Using "LabelEncoder" package, we recode the column (name as Sentiment) in numeric form (like 2: Positive, 0: Negative, 1: Neutral). Using Scikit-Learn module named "countvectorizer", we convert the text data into numeric format and prepare the matrix of tokens count.

Finally the final dataset is ready for machine learning model to perform Sentiment Analysis.

### **3.3 Data Splitting**

We need to split a dataset into train and test sets to evaluate how well our machine learning model performs. The train set is used to fit the model, and the statistics of the train set are known. The second set is called the test data set, this set is solely used to check the accuracy of the model.

Here we split our dataset into 70% for train the model and 30% for test the model.

### **3.4 Creating ML Model**

Using the train dataset which is splitting in previous step, here we created most popular machine learning algorithm, namely Naïve-Bayes(NB), Support Vector Machine(SVM), and Random Forest(RF) to perform the Sentiment analysis in better way.

After computing the accuracy of each model, We then select the best model which is giving the best result and which classifier is better in a specific Scenario.

### **3.5 Conclusion**

With the choice of best model which classified the sentiment on the YouTube comments about "Samsung Galaxy S23 Ultra" in better way, here we draw some conclusion of public sentiment on the brand image of upcoming smartphone "Samsung Galaxy S23 Ultra" and also make suggestions for modification in the upcoming versions of Samsung Galaxy S23 Ultra to it's developers based on the negative comments received on that video.

### **3.6 framework of python and deploy it on Heroku platform**

At last,we created a web application on Heroku platform using flask which is a framework of python. The web application serves as user interface where user will search and obtain the results.

## **4 ML Algorithms used on Sentiment Analysis:**

Sentiment analysis is the process of analyzing text with the help of machine learning to identify the polarity of text. Sentiment analysis is the method used to evaluate a sentence or a word on the basis of sentiment. There are mainly two approaches used for sentiment analysis. One method is to use the dictionary where each word is represented by a numerical value as polarity. The next method is machine learning, where statistical methods are employed to find out the vectorized value of a word via word embedding. After that, the machine learning algorithm is trained using the digitized value of a word or a sentence. There are multiple machine learning algorithms used for sentiment analysis like Support Vector Machine (SVM), Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), Random Forest, Naïve Bayes, and Long Short-Term Memory (LSTM). In this study, we have performed the sentiment classification using Naïve Bayes, Random Forest and Support Vector Machine (SVM) to classify the comments as positive, negative and neutral sentiment.

### **4.1 Naïve Bayes classifier:**

Naive Bayes is a powerful algorithm that is used for text data analysis and with problems with multiple classes. To understand Naive Bayes theorem's working, it is important to understand the Bayes theorem concept at first. Bayes theorem, formulated by Thomas Bayes,

calculates the probability of an event occurring based on the prior knowledge of conditions related to an event. It is based on the following formula:

$$\mathbf{P}(\mathbf{A} \mid \mathbf{B}) = \frac{\mathbf{P}(\mathbf{A}) * \mathbf{P}(\mathbf{B} \mid \mathbf{A})}{\mathbf{P}(\mathbf{B})}$$

Where we are calculating the probability of class A when predictor B is already provided.

$\mathbf{P}(\mathbf{B})$  = prior probability of B

$\mathbf{P}(\mathbf{A})$  = prior probability of class A

$\mathbf{P}(\mathbf{B} \mid \mathbf{A})$  = occurrence of predictor B given class A probability

This formula helps in calculating the probability of the tags in the text.

Naive Bayes is a probabilistic classifier, meaning that for a document  $d$ , out of all classes  $c \in C$  the classifier returns the class  $\hat{c}$  which has the maximum posterior probability given the document. we use the hat notation  $\hat{\phantom{x}}$  to mean “our estimate of the correct class”.

$$\begin{aligned}\hat{c} &= \arg \max_{c \in C} \mathbf{P}(c \mid d) \\ &= \arg \max_{c \in C} \frac{\mathbf{P}(d \mid c) * \mathbf{P}(c)}{\mathbf{P}(d)}\end{aligned}$$

Since  $\mathbf{P}(d)$  doesn't change for each class; we are always asking about the most likely class for the same document  $d$ , which must have the same probability  $\mathbf{P}(d)$ . Thus, we can choose the class that maximizes this simpler formula:

$$\hat{c} = \arg \max_{c \in C} \mathbf{P}(d \mid c) * \mathbf{P}(c)$$

**To return to classification:** we compute the most probable class  $\hat{c}$  given some document  $d$  by choosing the class which has the highest

product of two probabilities: the **prior probability** of the class  $P(c)$  and the **likelihood** of the document  $P(d | c)$ :

$$\hat{c} = \arg \max_{c \in C} P(d | c) * P(c)$$

where  $P(d | c)$  is the **likelihood** and  $P(c)$  is the **prior probability** . Without loss of generalization, we can represent a document  $d$  as a set of features  $f_1, f_2, \dots, f_n$ :

$$\hat{c} = \arg \max_{c \in C} P(f_1, f_2, \dots, f_n | c) * P(c)$$

Unfortunately, this equation is still too hard to compute directly: without some simplifying assumptions, estimating the probability of every possible combination of features (for example, every possible set of words and positions) would require huge numbers of parameters and impossibly large training sets. Naive Bayes classifiers therefore make two simplifying assumptions.

**First:** we assume the position word doesn't matter on classification whether it occurs as the 1st, 20th, or last word in the document. Thus we assume  $f_1, f_2, \dots, f_n$  that the features only encode word identity and not position.

**Second:** the conditional independence assumption that the probabilities  $P(f_i | c)$  are independent given the class  $c$  and hence can be 'naively' multiplied as follows:

$$P(f_1, f_2, \dots, f_n | c) = P(f_1 | c) * P(f_2 | c) * \dots * P(f_n | c)$$

The final equation for the class chosen by a naive Bayes classifier is thus:

$$c_{NB} = \arg \max_{c \in C} P(c) \prod_{f \in F} P(f | c)$$

**To apply the naive Bayes classifier to text,**

we consider words as the features of the document and we need to consider word positions, by simply walking an index through every word position in the document:

positions  $\leftarrow$  all word positions in test document

$$\mathbf{c}_{\text{NB}} = \arg \max_{\mathbf{c} \in \mathbf{C}} \mathbf{P}(\mathbf{c}) \prod_{\mathbf{i} \in \text{positions}} \mathbf{P}(\mathbf{w}_{\mathbf{i}} \mid \mathbf{c})$$

where  $\mathbf{w}_{\mathbf{i}}$  is the word in the  $\mathbf{i}$ -th position in the text document.

Now, we estimate  $\mathbf{P}(\mathbf{c})$  and  $\mathbf{P}(\mathbf{w}_{\mathbf{i}} \mid \mathbf{c})$ . Let's first consider the maximum likelihood estimate. We'll simply use the frequencies in the data. For the class prior  $\mathbf{P}(\mathbf{c})$  we ask what percentage of the documents in our training set are in each class  $\mathbf{c}$ . Let  $N_{\mathbf{c}}$  be the number of documents in our training data with class  $\mathbf{c}$  and  $N_{\text{doc}}$  be the total number of documents. Then:

$$\hat{P}(\mathbf{c}) = \frac{N_{\mathbf{c}}}{N_{\text{doc}}}$$

$$\hat{P}(w_i \mid \mathbf{c}) = \frac{\text{count}(w_i, \mathbf{c}) + 1}{\sum_{w \in V} \text{count}(w, \mathbf{c}) + |V|}$$

where the vocabulary  $V$  consists of the union of all the word types in all classes, not just the words in one class  $\mathbf{c}$ .

## 4.2 Support Vector Machine(SVM):

SVM is a supervised machine learning algorithm that helps in both classification and regression problem statements. It commonly used for classification problem. It tries to find an optimal decision boundary between different classes. SVM does complex data transformations depending on the selected kernel function, and based on those transformations, it aims to maximize the separation boundaries between our data points. If we choose the linear kernel function then best decision boundary is called **hyperplane**, otherwise we have non-linear

decision boundary. SVM chooses the extreme points/vectors that help in creating the decision boundary. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.

Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:

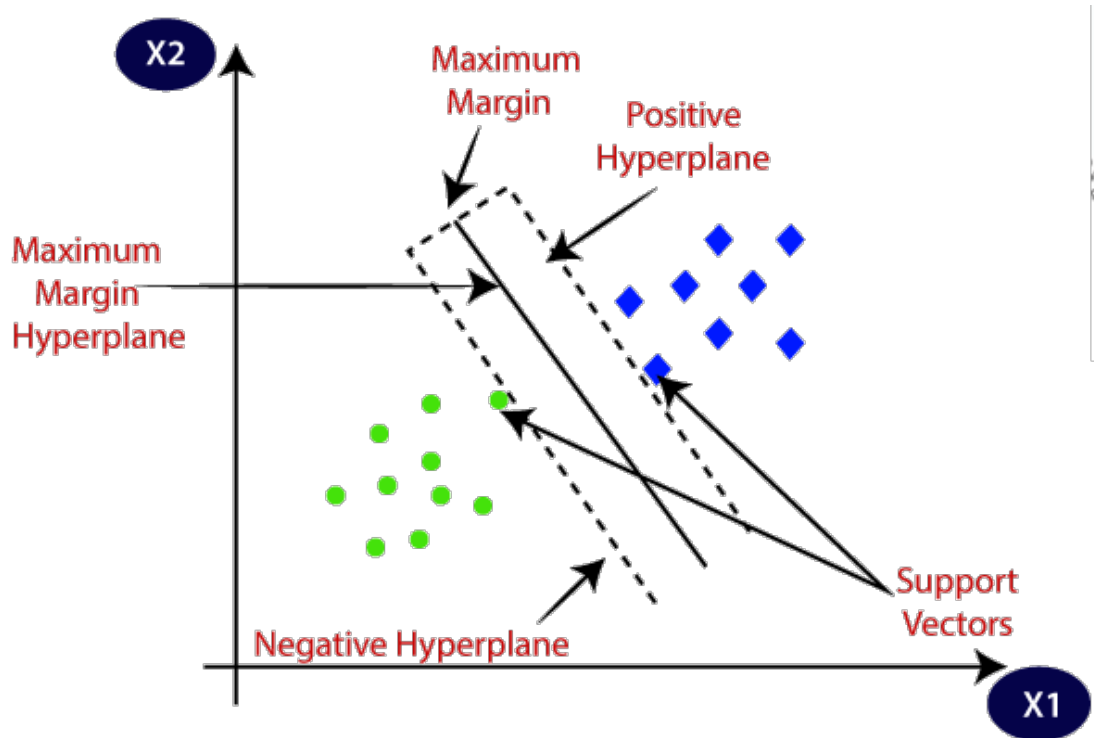


Figure 1.2:SVM classifier with linear kernel(hyperplane).

# Nonlinear decision boundary

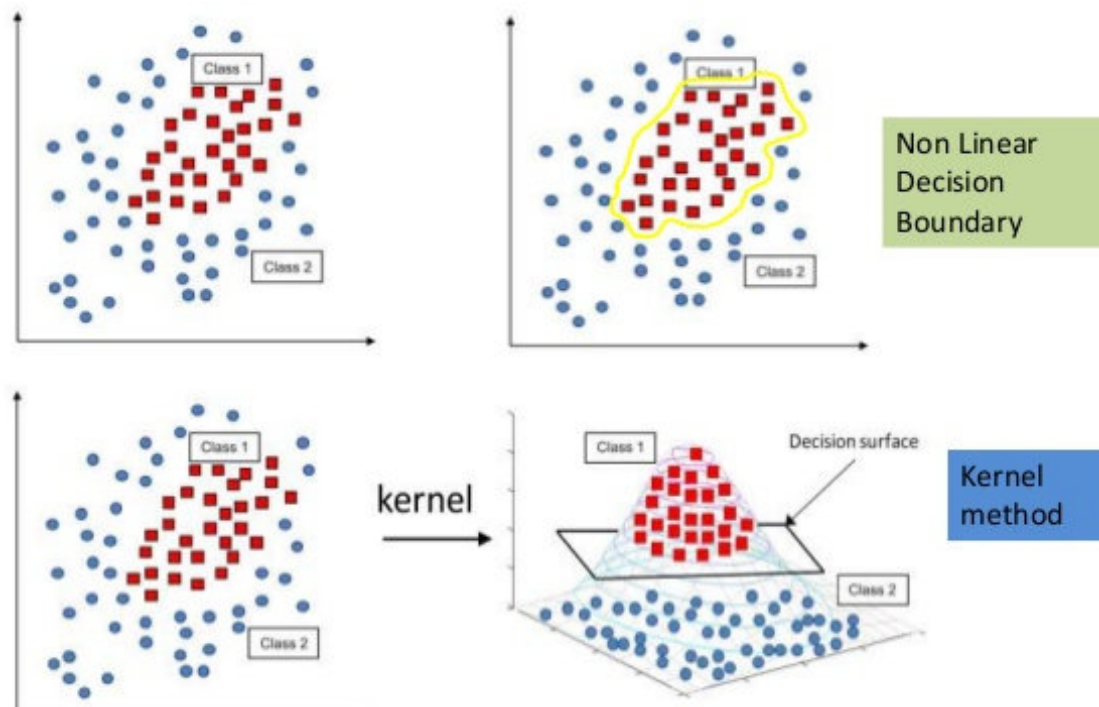


Figure 1.2:SVM classifier with Non-linear kernel.

## 4.3 Random Forest:

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of **ensemble learning**, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

”Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. Instead of relying on



one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

The below diagram explains the working of the Random Forest algorithm:

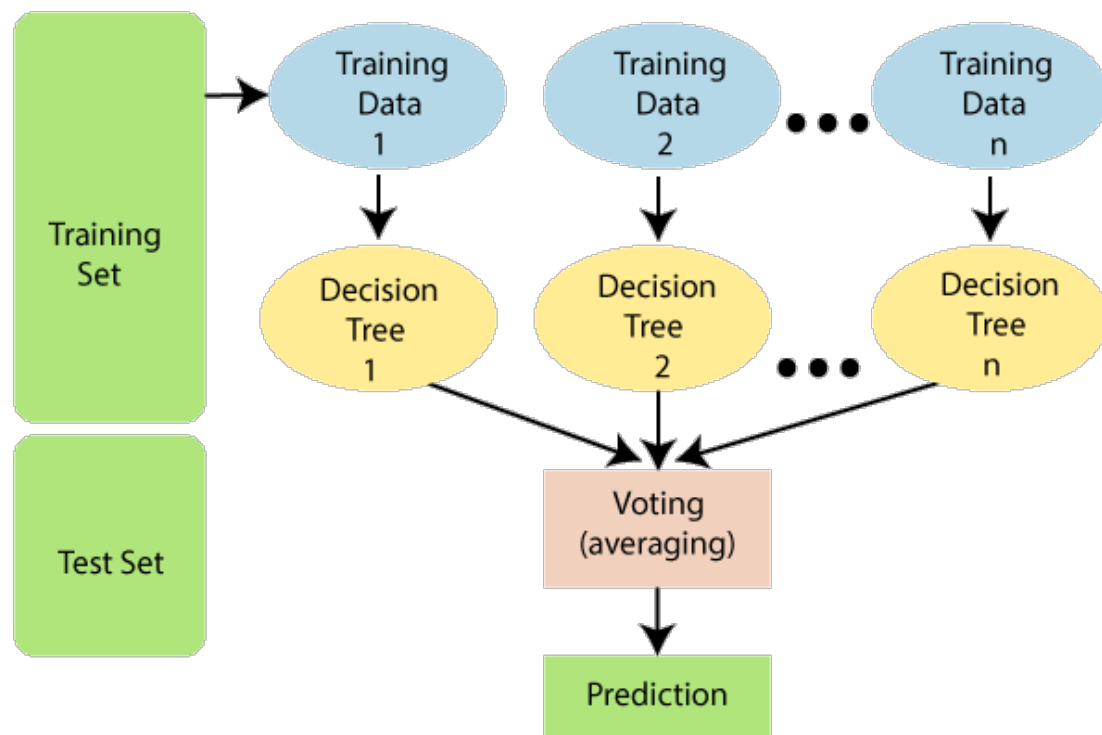


Figure 1.3: Random Forest algorithm diagram.

#### 4.3.1 Assumptions for Random Forest:

Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees

predict the correct output. Therefore, below are two assumptions for a better Random forest classifier:

- There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.
- The predictions from each tree must have very low correlations.

#### **4.3.2 Why use Random Forest?**

Below are some points that explain why we should use the Random Forest algorithm:

- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.

#### **4.3.3 Random Forest Algorithm:**

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps and diagram:

**Step-1:** Select random K data points from the training set.

**Step-2:** Build the decision trees associated with the selected data points (Subsets).

**Step-3:** Choose the number N for decision trees that you want to build.

**Step-4:** Repeat Step 1 & 2.

**Step-5:** For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes. **Example:** Suppose there is a dataset that contains multiple fruit images. So, this dataset is given to the Random forest classifier. The dataset is divided into subsets and given to each decision tree. During the training phase, each decision tree produces a prediction result, and when a new data point occurs, then based on the majority of results, the Random Forest classifier predicts the final decision. Consider the below image:

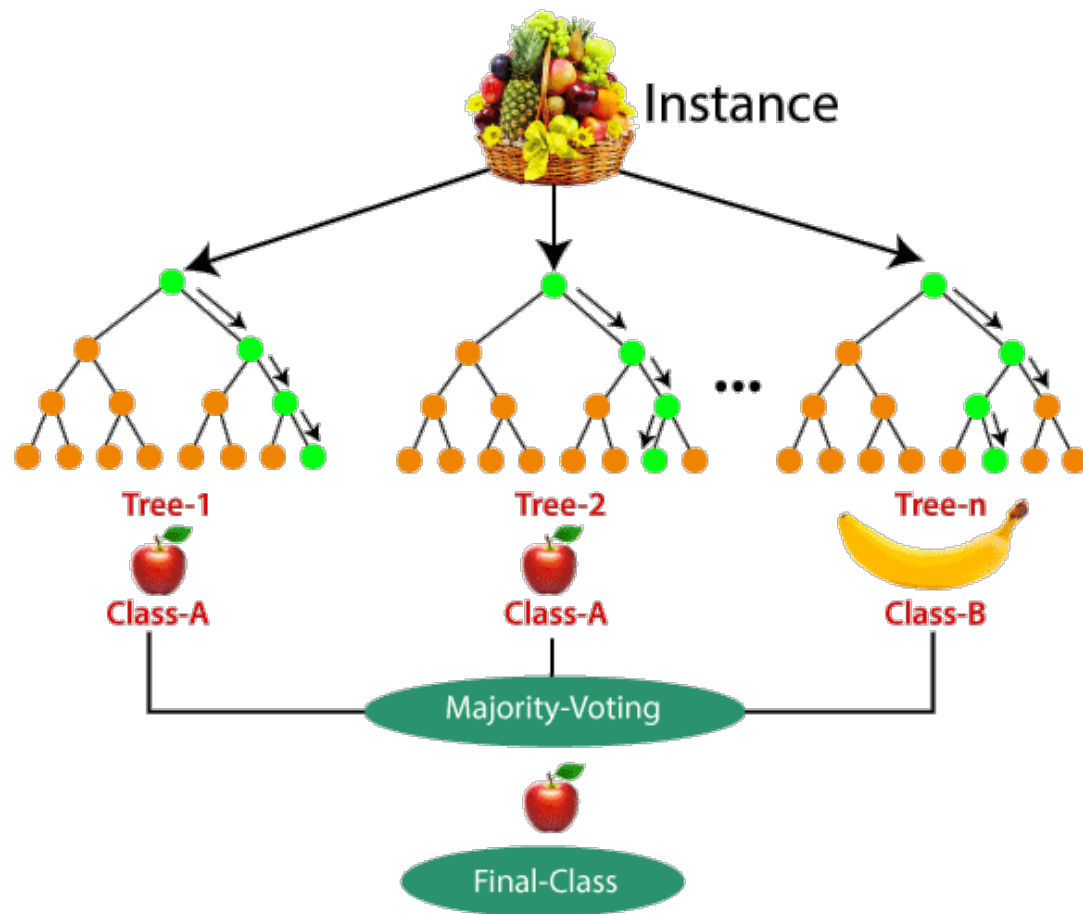


Figure 1.3.3: Random Forest algorithm diagram example.

## 4.4 Model Performance Checking:

Now we calculate the model performance score on these three performed classification model and using this score we will choose the best performing model that has highest score.

To check model efficiency on test data, we can consider following performance metric.

- **Accuracy:**

Accuracy is used in classification problems to tell the percentage of correct predictions made by a model. Accuracy score in machine learning is an evaluation metric that measures the number of correct predictions made by a model in relation to the total number of predictions made. We calculate it by dividing the number of correct predictions by the total number of predictions.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

Accuracy is simple to calculate but has its own disadvantage.

- If the data set is highly imbalanced, and the model classifies all the data points as the majority class data points, the accuracy will be high. This makes accuracy not a reliable performance metric for imbalanced data.

- **Confusion Matrix:**

Confusion Matrix is a summary of predicted results in specific table layout that allows visualization of the performance measure of the machine learning model for a binary classification problem (2 classes) or multi-class classification problem (more than 2 classes)

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

- TP means **True Positive**. It can be interpreted as the model predicted positive class and it is True.
- FP means **False Positive**. It can be interpreted as the model predicted positive class but it is False.
- FN means **False Negative**. It can be interpreted as the model predicted negative class but it is False.
- TN means **True Negative**. It can be interpreted as the model predicted negative class and it is True.

- **Precision:**

The precision metric is used to overcome the limitation of Accuracy. The precision determines the proportion of positive prediction that was actually correct. It can be calculated as the True Positive or predictions that are actually true to the total positive predictions (True Positive and False Positive)

$$\textbf{Precision} = \frac{TP}{(TP + FP)}$$

- **Recall/Sensitivity:**

It is also similar to the Precision metric; however, it aims to calculate the proportion of actual positive that was identified incorrectly. It can be calculated as True Positive or predictions that are actually true to the total number of positives, either correctly predicted as positive or incorrectly predicted as negative (true Positive and false negative).

$$\textbf{Recall} = \frac{TP}{TP + FN}$$

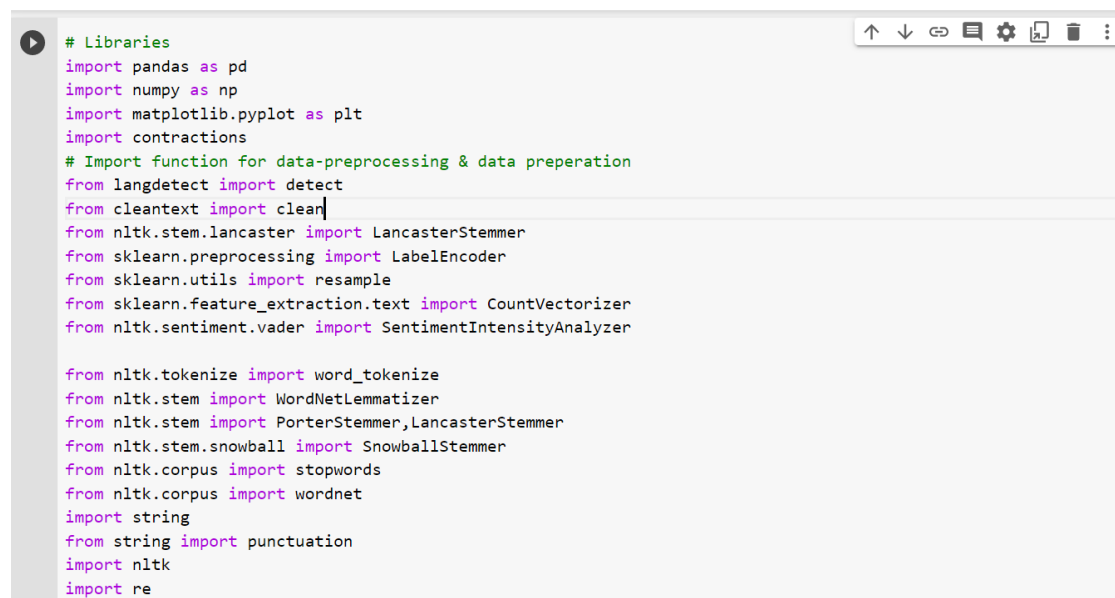
**F1 Score:**

It is very popular metric to evaluate model efficiency for imbalanced data. It is calculated with the help of Precision and Recall. the F1 Score can be calculated as the harmonic mean of both precision and Recall, assigning equal weight to each of them.

$$F1 - score = 2 * \frac{precision * recall}{precision + recall}$$

## 5 Performing Sentiment Analysis of YouTube comments on Samsung Galaxy S23 Ultra's official trailer:

Python Packages:

A screenshot of a code editor window with a light gray background. The code is written in Python and is color-coded. It starts with a comment '# Libraries' in green. The imports include pandas as pd, numpy as np, matplotlib.pyplot as plt, and contractions. A green comment line reads '# Import function for data-preprocessing & data preperation'. The code then imports detect from langdetect, clean from cleantext, LancasterStemmer from nltk.stem.lancaster, LabelEncoder from sklearn.preprocessing, resample from sklearn.utils, CountVectorizer from sklearn.feature\_extraction.text, and SentimentIntensityAnalyzer from nltk.sentiment.vader. A second set of imports includes word\_tokenize from nltk.tokenize, WordNetLemmatizer from nltk.stem, PorterStemmer and LancasterStemmer from nltk.stem, SnowballStemmer from nltk.stem.snowball, stopwords from nltk.corpus, and wordnet from nltk.corpus. Finally, it imports string, punctuation from string, and nltk and re at the bottom. The editor has a toolbar on the right with icons for undo, redo, search, settings, and other standard functions.

```
# Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import contractions

# Import function for data-preprocessing & data preperation
from langdetect import detect
from cleantext import clean

from nltk.stem.lancaster import LancasterStemmer
from sklearn.preprocessing import LabelEncoder
from sklearn.utils import resample
from sklearn.feature_extraction.text import CountVectorizer
from nltk.sentiment.vader import SentimentIntensityAnalyzer

from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
from nltk.stem import PorterStemmer, LancasterStemmer
from nltk.stem.snowball import SnowballStemmer
from nltk.corpus import stopwords
from nltk.corpus import wordnet
import string
from string import punctuation
import nltk
import re
```

### 5.1 Data collect from YouTube:

As we discussed before, instead of YouTube API we collect the YouTube comments on Samsung Galaxy S23 Ultra's official trailer using python package such as "bot-studio" and "youtube-comment-scraper-python".

```

pip install bot-studio # Installing bot-studio

from bot_studio import * # Importing all module from bot-studio

pip install youtube-comment-scraper-python # Installing youtube-comment-scraper-python

from youtube_comment_scraper_python import * # Importing all module from youtube-comment-scraper-python

import pandas as pd # Importing pandas to work on dataset.

link = input("Youtube links: ")
saved = input("Output name: ")
youtube.open(link)

response = youtube.video_comments()
data = response['body']

df = pd.DataFrame(data)
df.to_csv(saved)

Youtube links: https://www.youtube.com/watch?v=BSYsXVFzmKA
Output name: output.csv
Progress: 100% 200.0/200 [00:24<00:00, 8.15it/s]

```

Using the above codes, we execute scrolling of page 40 times and collect about 900 comments and saved dataset as "csv" file in our Desktop with name "output.csv"

	Unnamed: 0	Comment	Likes	Time	UserLink	user
0	0	Play on with a super powerful processor and s...	852	2 months ago	https://www.youtube.com/channel/UCWwgaK7x0_FR1...	NaN
1	1	Ok this is awesome. Im so happy that Samsung i...	1.1K	2 months ago	NaN	NaN
2	2	Ever since the S22 Ultra, I've been the bigges...	951	2 months ago	NaN	NaN
3	3	By far the most creative and captivating intro...	94	2 months ago	NaN	NaN
4	4	Probably one of the best smartphone trailers E...	283	2 months ago	NaN	NaN

Comment	Li
0 Play on with a super powerful processor and snap away in ultra-high resolution. What epic moments will you share? Learn more: http://smsng.co/S23Ultra_Intro_ytp	
1 Ok this is awesome. Im so happy that Samsung is upgrading it's night photography and giving some focus to astrophotography as well. One of my favorite things to do is mobile astrophotogra	1.
2 Ever since the S22 Ultra, I've been the biggest fan of your smartphones! This year is going to be even better, and now with more steady video as well as a vastly improved night mode, I'm sure	
3 By far the most creative and captivating introduction film of any product.	
4 Probably one of the best smartphone trailers EVER!, the S23 is going to be huge this year!	
5 This is the best Samsung commercial in decades! Makes me really want to get the S23. Maybe I will	1.
6 The introduction film is impecceable! Looooooved watching every second of it.	
7 Nostalgia to the times when smartphone introductions are fun like this	3.
8 This is actually such an elaborate and fun introduction film. Nice job. My s23 Ultra is already pre-ordered!	
9 I have to admit that this focus on the camera including Astro photography and low light seems pretty great.	
10 Never seen Samsung pull out an epic commercial better than this	
11 Hmmâ€¦ still not convincing me enough to move from 14 pro max.	
12 I must confess I love this commercial, beautifully done and fun too	
13 Samsung Official Introductions are not as good as they were earlier	
14 Great one! Both intro film and the product looks absolutely promising.	
15 Samsung always innovates endlessly, I pray everyone can buy Samsung S23 Ultra. Love it	
16 The dumb equipment of 700 performance better	
17 I must say Samsung has the best video introduction of all time. Kind of disappointed with my trade in value for my s22 ultra, but Iâ€™ll hold onto it until next year	
18 I got mine yesterday and as Samsung stated, "Ultra" is the best of the best and the S23 Ultra delivers. The device is so responsive that its unbelievable. My Note 20 Ultra wasnt this fast at all.	
19 And I might upgrade from S22 Ultra to S23 Ultra just for that astrophotography.	



## 5.2 Data Pre-processing:

### 5.2.1 Features selection:

Our dataset has various columns(features) like 'Unnamed', 'Comment', 'Likes', 'Time', 'UserLink', 'user'. But in our analysis part only 'Comment' features is required to perform Sentiment Analysis. So we will drop unnecessary columns from our original dataset using the following commands.

```
data = pd.read_csv('Sentiment_data.csv')
data1=data.drop(['Unnamed: 0', 'Likes', 'Time', 'user', 'UserLink'],axis=1)
data1
```

index	Comment
0	Play on with a super powerful processor and snap away in ultra-high resolution. What epic moments will you share?Learn more: <a href="http://smsng.co/S23Ultra_Intro_ytp">http://smsng.co/S23Ultra_Intro_ytp</a>
1	Ok this is awesome. Im so happy that Samsung is upgrading it's night photography and giving some focus to astrophotography as well. One of my favorite things to do is mobile astrophotography and this is looking like a serious upgrade for me. Thank you Samsung, will definitely be upgrading from my 5 year old OnePlus this year!
2	Ever since the S22 Ultra, I've been the biggest fan of your smartphones! This year is going to be even better, and now with more steady video as well as a vastly improved night mode, I sure we will even beat Apple's recording capabilities!
3	By far the most creative and captivating introduction film of any product. Way to go, Samsung!
4	Probably one of the best smartphone trailers EVER!, the S23 is going to be huge this year!
5	This is the best Samsung commercial in decades! Makes me really want to get the S23. Maybe I will UPDATE: The retailers are borderline scalpers. The phone costs double the price where I live in the Caribbean. I'll have to sit this one out. Ad is still sick
6	The introduction film is impecceable! Looooooved watching every second of it.
7	Nostalgia to the times when smartphone introductions are fun like this Edit: thank you for the likes
8	This is actually such an elaborate and fun introduction film. Nice job. My s23 Ultra is already pre-ordered!
9	I have to admit that this focus on the camera including Astro photography and low light seems pretty great.
10	Never seen Samsung pull out an epic commercial better than this

### 5.2.2 Deleting Non-English comments:

There are some non-English comments in our dataset which has to delete because we are performing sentiment analysis on English comments(There is a multilingual sentiment analysis where more than one language are accepted but here we are bounded to English comments). At the below,we show some non-English comments as:

	B
11	Meu Deus que perfeito, amei de mais! i,
12	I must confess I love this commercial, beautifully done and fun too
13	Que introdu��o linda faz a gente querer assistir at�� o ��ltimo segundo , parab��ns samsung vcs se superaram
14	Great one! Both intro film and the product looks absolutely promising.
15	Samsung always innovates endlessly, I pray everyone can buy Samsung S23 Ultra. Love it
16	As marcas sempre mexem na mem��ria, processador e c��mera, ca��ram numa grande mesmice. O pr��ximo s24, s25, s26 ��! v��o vir com os mesmos upgrades de sempre. Olha por exemp
17	I must say Samsung has the best video introduction of all time. Kind of disappointed with my trade in value for my s22 ultra, but I��ll hold unto it until next year
18	I got mine yesterday and as Samsung stated, "Ultra" is the best of the best and the S23 Ultra delivers. The device is so responsive that its unbelievable. My Note 20 Ultra wasnt this fast at all.
19	And I might upgrade from S22 Ultra to S23 Ultra just for that astrophotography.
20	It was fun watching it.Awesome introduction i,
21	As someone who has a S9 and loved taking photos on my phone. This is what I'll upgrade to
22	Been so long since i saw a genuinely fun intro video for a smartphone launch. Way to go Samsung. I might upgrade from my old one plus this year!
23	I always wait for new Samsung introduction videos they are always fun to watch love it as always i,
24	Great phone as always. Love the design. I wish if I had money to own this beauty
25	Com certeza absoluta, vou passar meu 22 ultra e peg��i-lo... Est��i melhor ainda i,
26	I haven't been this excited for a new phone in a while!

The removing non-English comments is a complete case analysis, so we selected those comments and removed from our dataset and we will continue with this dataset(containing only English comments).

### 5.2.3 Data Labelling:

We will perform Sentiment analysis to find the public sentiment about the flagship smartphone("Samsung Galaxy S23 Ultra"). Here we will use supervised Machine learning(ML) classification algorithm to classify the public comments as positive, neutral or negative feedback. For this we need to label the original comments as positive, negative and neutral based on the polarity score of each comments. Here we used python module such as "SentimentIntensityAnalyzer()" which is a part of "NLTK" and "vaderSentiment" python packages to find the polarity score of a text. If the polarity score is greater than equals to 0.5, it labelled as "positive" and if the polarity score is less than equals to -0.5, it labelled as "negative", otherwise it assigned as "neutral".

```

nltk.download("vader_lexicon")
sentiments=SentimentIntensityAnalyzer()
data["Positive"]=[sentiments.polarity_scores(i)["pos"] for i in data["Comment"]]
data["Negative"]=[sentiments.polarity_scores(i)["neg"] for i in data["Comment"]]
data["Neutral"]=[sentiments.polarity_scores(i)["neu"] for i in data["Comment"]]
data["Compound"]=[sentiments.polarity_scores(i)["compound"] for i in data["Comment"]]
score=data["Compound"].values
sentiment=[]
for i in score:
    if i>=0.5:
        sentiment.append("Positive")
    elif i<=-0.5:
        sentiment.append("Negative")
    else:
        sentiment.append("Neutral")
data["Sentiment"]=sentiment
data_label=data.drop(["Positive","Negative","Neutral","Compound"],axis=1)
data_label.head()  ## This is our Final data on which we will apply Sentiment Analysis

```

Below we have attached first and last 5 rows of labelled dataset.

index	Comment	Sentiment
0	Play on with a super powerful processor and snap away in ultra-high resolution. What epic moments will you share?Learn more: <a href="http://smsng.co/S23Ultra_Intro_ytp">http://smsng.co/S23Ultra_Intro_ytp</a>	Positive
1	Ok this is awesome. Im so happy that Samsung is upgrading it's night photography and giving some focus to astrophotography as well. One of my favorite things to do is mobile astrophotography and this is looking like a serious upgrade for me. Thank you Samsung, will definitely be upgrading from my 5 year old OnePlus this year!	Positive
2	Ever since the S22 Ultra, I've been the biggest fan of your smartphones! This year is going to be even better, and now with more steady video as well as a vastly improved night mode, I'm sure we will even beat Apple's recording capabilities!	Positive
3	By far the most creative and captivating introduction film of any product. Way to go, Samsung!	Positive
4	Probably one of the best smartphone trailers EVER!, the S23 is going to be huge this year!	Positive

index	Comment	Sentiment
799	Samsung is always amazing	Positive
800	But i phone is dream for everyone	Neutral
801	Whatever they produce, after 2 years you have to visit service center to change display and they will charge 1/2 of the original phone price. Enjoy	Positive
802	I would hate to drop this phone on the floor	Negative
803	My Pone Is Samsung A10s	Neutral

#### 5.2.4 Data cleaning & Transformation:

Data cleaning and transformation is one of the important part of pre-processing because the machine learning model will not perform

on raw dataset. Here we cleaned each comment in our dataset step by step as follows:

### **1.Lower Casing the Data:**

Another problem with sentiment analysis are capital letters. Imagine the word “us”. It could be:

- A pronoun representing “we” on the sentence;
- The country “USA”.

Now, imagine that we have a review from someone really angry. In order to represent their anger, the person decided to write everything in capital letters. In this context, this style choice adds information to the analysis, since it represents anger and frustration.

Again, in a some text words like Book and book mean the same but when not converted to the lower case those two are represented as two different words in the vector space model (resulting in more dimensions).

Thus it is better to convert the text in lowercase.

### **2.Punctuation Removing:**

The second most common text processing technique is removing punctuations from the textual data. The punctuation removal process will help to treat each text equally. For example, the word data and data! are treated equally after the process of removal of punctuations. We need to take care of the text while removing the punctuation because the contraction words will not have any meaning after the punctuation removal process. Such as ‘don’t’ will convert to ‘dont’ or ‘don t’ depending upon what you set in the parameter. So before removing punctuations, we should be removed the contractions such as ‘don’t’ to ‘do not’.

### **3.Removing Extra Space:**

Well, removing the extra space is good as it doesn't store extra memory and even we can see the data clearly.

#### **4.Removing new lines:**

Sometimes there are unnecessary new lines in the text which holds extra memory and it effects to see the whole text clearly, so it is better to remove from text.

#### **5.Removing references,hashtagsspecial characters:**

References,hashtags special characters in the text do not contribute much to find the sentiment along the text, so it's good to remove them.

#### **6.Removing Emojis:**

Growing users of the audience on the social media platforms, well there is a significant explosion of usage of emojis in day-to-day life. Well, when we are performing text analysis in some cases removal of emojis is the correct way as sometimes they don't hold any information.

**7.Word tokenization:** Tokenization is the process of breaking down the given text in natural language processing into the smallest unit in a sentence called a token.When we will perform the "Lemmatization"(next process) , it is required to split the text into words and that time Word tokenization process will help us.

#### **8.Stemming & Lemmatization:**

Both stemming lemmatization methods are used to converts the word in its root/base form, for instance "better" to "good". "Stemming" is preferred when the meaning of the word is not important for analysis,example: Spam Detection. where "Lemmatization" would be recommended when the meaning of the word is important for analysis,example: Sentiment Analysis. Thus we apply "Lemmatization" in our context to find the base form of each word.

We used the following python codes to clean & transform the dataset.

```

!pip install contractions # Installing contractions packages to remove contraction from text.
import contractions
lzh = WordNetLemmatizer()
def text_processing(text):
    # convert text into lowercase
    text = text.lower()
    #remove unnecessary URL link
    text = re.sub(r"http\S+", "", text)
    # remove new line characters in text
    text = re.sub(r'\n', ' ', text)
    # remove contractions
    text=contractions.fix(text)
    # remove punctuations from text
    text = re.sub('[%s]' % re.escape(punctuation), "", text)
    # remove references and hashtags from text
    text = re.sub("^a-zA-Z0-9$","", text)
    # remove multiple spaces from text
    text = re.sub(r'\s+', ' ', text, flags=re.I)
    # remove special characters from text
    text = re.sub(r'\W', ' ', text)
    #text = ' '.join([word for word in word_tokenize(text) if word not in stop_words])
    # lemmatizer using WordNetLemmatizer from nltk package
    text=' '.join([lzh.lemmatize(word) for word in word_tokenize(text)])
    return text

```

Before data pre-processing:

index	Comment	Sentiment
0	Play on with a super powerful processor and snap away in ultra-high resolution. What epic moments will you share?Learn more: <a href="http://smsng.co/S23Ultra_Intro_yp">http://smsng.co/S23Ultra_Intro_yp</a>	Positive
1	Ok this is awesome. Im so happy that Samsung is upgrading it's night photography and giving some focus to astrophotography as well. One of my favorite things to do is mobile astrophotography and this is looking like a serious upgrade for me. Thank you Samsung, will definitely be upgrading from my 5 year old OnePlus this year!	Positive
2	Ever since the S22 Ultra, I've been the biggest fan of your smartphones! This year is going to be even better, and now with more steady video as well as a vastly improved night mode, I'm sure we will even beat Apple's recording capabilities!	Positive
3	By far the most creative and captivating introduction film of any product. Way to go, Samsung!	Positive
4	Probably one of the best smartphone trailers EVER!, the S23 is going to be huge this year!	Positive
5	This is the best Samsung commercial in decades! Makes me really want to get the S23. Maybe I will UPDATE: The retailers are borderline scalpers. The phone costs double the price where I live in the Caribbean. I'll have to sit this one out. Ad is still sick	Positive
6	The introduction film is impeccable! Looooooved watching every second of it.	Neutral
7	Nostalgia to the times when smartphone introductions are fun like this Edit: thank you for the likes	Positive
8	This is actually such an elaborate and fun introduction film. Nice job. My s23 Ultra is already pre-ordered!	Positive
9	I have to admit that this focus on the camera including Astro photography and low light seems pretty great.	Positive
10	Never seen Samsung pull out an epic commercial better than this	Positive
11	Hmmm... still not convincing me enough to move from 14 pro max.	Negative
12	I must confess I love this commercial, beautifully done and fun too	Positive
13	Samsung Official Introductions are not as good as they were earlier	Negative
14	Great one! Both intro film and the product looks absolutely promising.	Positive
15	Samsung always innovates endlessly, I pray everyone can buy Samsung S23 Ultra. Love it	Positive

After data pre-processing:

index	Comment	Sentiment
0	play on with a super powerful processor and snap away in ultrahigh resolution what epic moment will you sharelearn more	Positive
1	ok this is awesome i am so happy that samsung is upgrading it is night photography and giving some focus to astrophotography a well one of my favorite thing to do is mobile astrophotography and this is looking like a serious upgrade for me thank you samsung will definitely be upgrading from my 5 year old oneplus this year	Positive
2	ever since the s22 ultra i have been the biggest fan of your smartphones this year is going to be even better and now with more steady video a well a a vastly improved night mode i am sure we will even beat apple recording capability	Positive
3	by far the most creative and captivating introduction film of any product way to go samsung	Positive
4	probably one of the best smartphone trailer ever the s23 is going to be huge this year	Positive
5	this is the best samsung commercial in decade make me really want to get the s23 maybe i will update the retailer are borderline scalper the phone cost double the price where i live in the caribbean i will have to sit this one out ad is still sick	Positive
6	the introduction film is impecceable loooooooved watching every second of it	Neutral
7	nostalgia to the time when smartphone introduction are fun like this edit thank you for the like	Positive
8	this is actually such an elaborate and fun introduction film nice job my s23 ultra is already preordered	Positive
9	i have to admit that this focus on the camera including astro photography and low light seems pretty great	Positive
10	never seen samsung pull out an epic commercial better than this	Positive
11	hmmm still not convincing me enough to move from 14 pro max	Negative
12	i must confess i love this commercial beautifully done and fun too	Positive
13	samsung official introduction are not a good a they were earlier	Negative
14	great one both intro film and the product look absolutely promising	Positive
15	samsung always innovates endlessly i pray everyone can buy samsung s23 ultra love it	Positive

### 5.2.5 Label Encoding:

The "Sentiment" column of the pre-processed dataset is a categorical variable having three levels as 'positive', 'negative' and 'neutral'. Later we will use machine learning algorithm to classify the levels for which it is required to encode the levels numerically which represents that categorical levels respectively. Here we used "LabelEncoder" package to recode the column(name as Sentiment) in numeric form like 2: Positive,0: Negative,1: Neutral.

```

le = LabelEncoder()
data_copy['Sentiment'] = le.fit_transform(data_copy['Sentiment'])
processed_data = {
    'Sentence': data_copy.Comment,
    'Sentiment': data_copy['Sentiment']
}

processed_data = pd.DataFrame(processed_data)
processed_data

```

Encoded dataset:

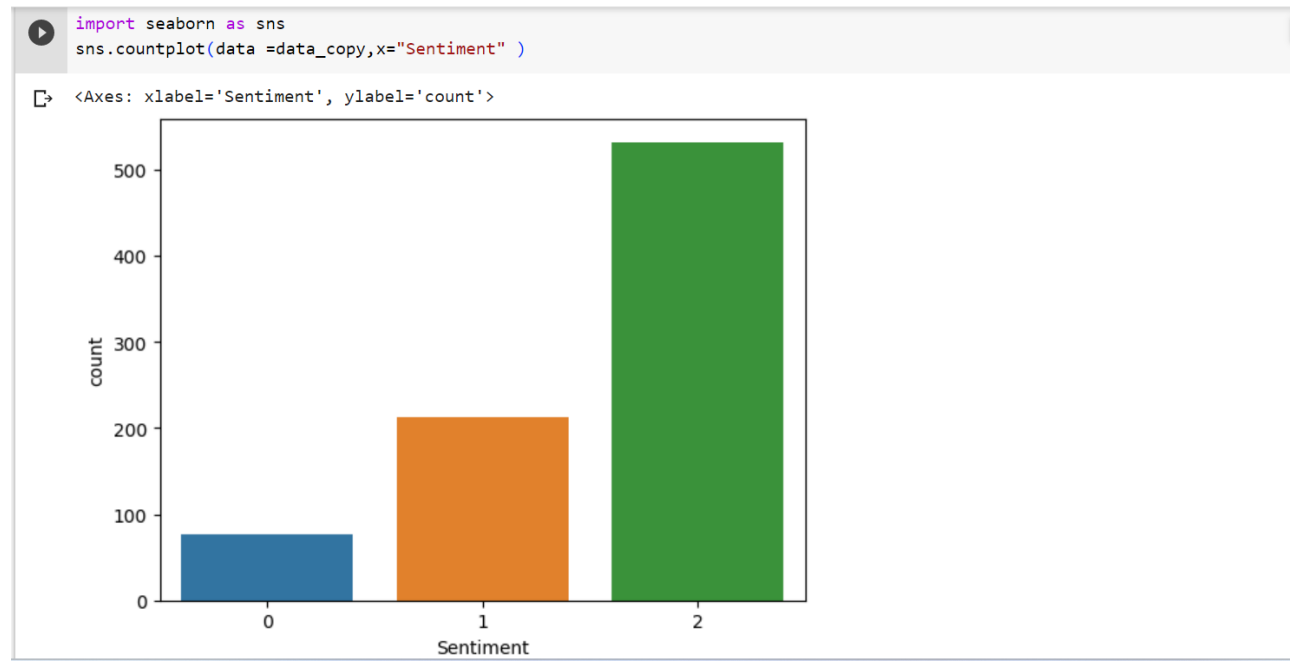
index	Sentence	Sentiment
0	play on with a super powerful processor and snap away in ultrahigh resolution what epic moment will you sharelearn more	2
1	ok this is awesome i am so happy that samsung is upgrading it is night photography and giving some focus to astrophotography a well one of my favorite thing to do is mobile astrophotography and this is looking like a serious upgrade for me thank you samsung will definitely be upgrading from my 5 year old oneplus this year	2
2	ever since the s22 ultra i have been the biggest fan of your smartphones this year is going to be even better and now with more steady video a well a a vastly improved night mode i am sure we will even beat apple recording capability	2
3	by far the most creative and captivating introduction film of any product way to go samsung	2
4	probably one of the best smartphone trailer ever the s23 is going to be huge this year	2
5	this is the best samsung commercial in decade make me really want to get the s23 maybe i will update the retailer are borderline scalper the phone cost double the price where i live in the caribbean i will have to sit this one out ad is still sick	2
6	the introduction film is impecceable loooooooved watching every second of it	1
7	nostalgia to the time when smartphone introduction are fun like this edit thank you for the like	2
8	this is actually such an elaborate and fun introduction film nice job my s23 ultra is already preordered	2
9	i have to admit that this focus on the camera including astro photography and low light seems pretty great	2
10	never seen samsung pull out an epic commercial better than this	2
11	hmmm still not convincing me enough to move from 14 pro max	0
12	i must confess i love this commercial beautifully done and fun too	2
13	samsung official introduction are not a good a they were earlier	0
14	great one both intro film and the product look absolutely promising	2
15	samsung always innovates endlessly i pray everyone can buy samsung s23 ultra love it	2

### 5.3 Balancing the dataset:

First we check the pre-processed dataset is balanced or unbalanced, if it is unbalanced dataset then we will convert into balanced dataset( by re-sampling method) as machine learning model will not perform well on unbalanced dataset. A classification data set with skewed class proportions is called unbalanced. Classes that make up a large



proportion of the data set are called majority classes. Those that make up a smaller proportion are minority classes.



The plots shows that the pre-processed dataset is unbalanced and hence re-sampling is required.

We used the following codes to execute re-sampling process.

```
[17] processed_data['Sentiment'].value_counts()
```

```
2    532
1    212
0     76
Name: Sentiment, dtype: int64
```

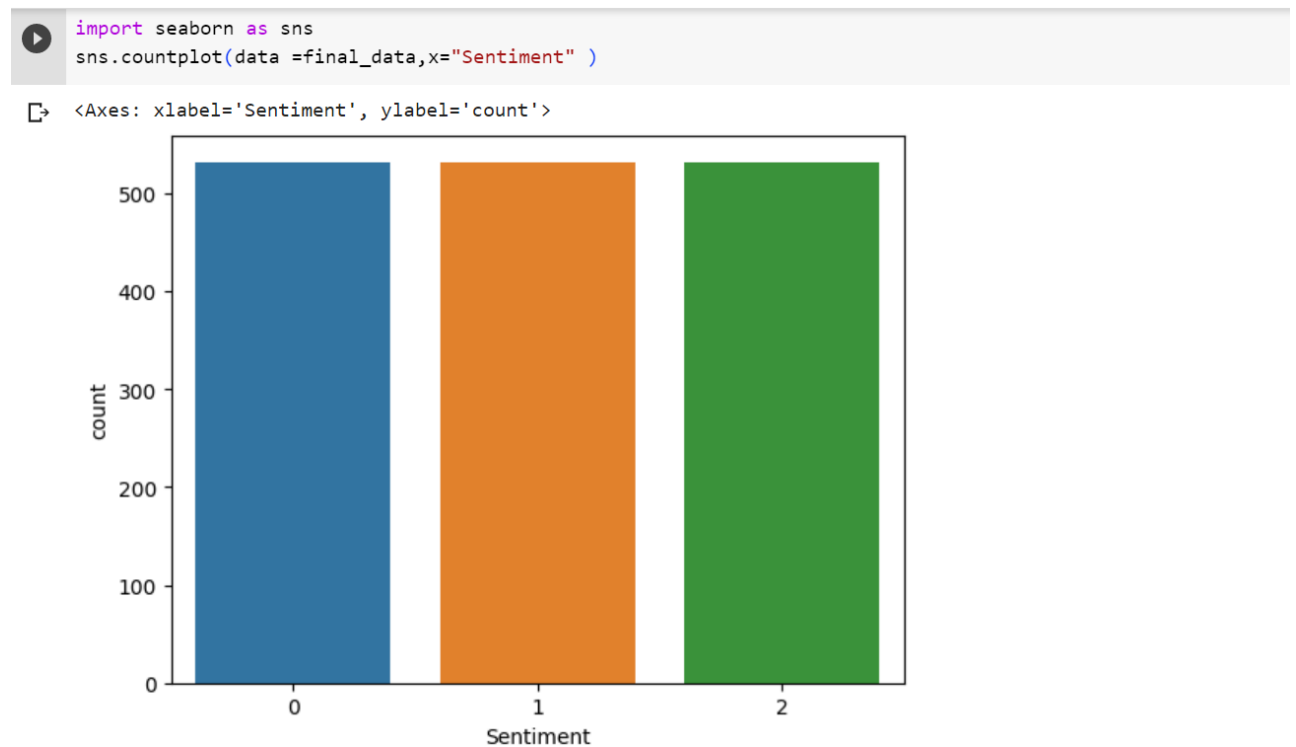
```
df_neutral = processed_data[(processed_data['Sentiment']==1)]
df_negative = processed_data[(processed_data['Sentiment']==0)]
df_positive = processed_data[(processed_data['Sentiment']==2)]

# upsample minority classes
df_negative_upsampled = resample(df_negative,
                                replace=True,
                                n_samples= 532,
                                random_state=42)

df_neutral_upsampled = resample(df_neutral,
                                replace=True,
                                n_samples= 532,
                                random_state=42)

# Concatenate the upsampled dataframes with the neutral dataframe
final_data = pd.concat([df_negative_upsampled,df_neutral_upsampled,df_positive])
```

The following plot shows that after re-sampling the pre-processed dataset becomes balanced.



finally, we got the dataset name as "final\_data" is the final data set on which we will performed machine learning algorithms.

## 5.4 Corpus & Count vectorization:

We want to build the classification model on the "final\_data" dataset where "Sentence" is the input features and "Sentiment" is the output(response). But the model will not work directly on text(comments) features,it needs to convert the each text into vector form by the frequencies of words.

"corpus" is the process to gather the all comments(pre-processed) into an array.

After creating "corpus", we will create an input matrix by "countvectorizer" where each row vectors represents the corresponding comments whose entries are the frequencies of the words in that comments

as well as the words in the "corpus". "countvectorizer" creates the columns (features in input matrix) with respect to words presents in the corpus.

We can consider all the words as a features in input matrix but it will effect to predict a new text because every time when new text will come ,the number of columns may increase in the input matrix and since our model has trained with old features matrix(less number of column) than new features matrix, our model will fails to predict new text's sentiment. Again, it is not necessary that all words present in a text will indicate the sentiment about some issue/case.

To overcome this problem, we set up a hyperparameter in "countvectorizer" function as "max\_features" whose value allows the function "countvectorizer" to create that number of features columns with respect to the words which are most likely repeated in corpus and we will give a value in this hyperparameter in such a way that it does not effect in Ml models as an "Overfitting" or "Underfitting".

Here we set up "max\_features"=1500 and creates an input features matrix as follows:

```
[153] from sklearn.feature_extraction.text import CountVectorizer
      cv = CountVectorizer(max_features=1500) # Count_vectorized function
      X = cv.fit_transform(corpus).toarray() # Input features matrix
      y = final_data.iloc[:, -1].values     # Response variable.
```

X

```
array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]])
```

```
[155] np.shape(X)

(1596, 1500)
```

## 5.5 Splitting & Machine learning models building:

To build a machine learning model, it is required to split the dataset into two parts as "train set" with 70% data and "test set" with 30% data.

### ▼ Data splitting:

```
[ ] from sklearn.naive_bayes import GaussianNB
    from sklearn.model_selection import train_test_split
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
```

Now we are ready to fit the machine learning classification model on "train set".

### 5.5.1 Gaussian Naive Bayes Classification:

#### ▼ Gaussian Naive Bayes:

```
classifier = GaussianNB()
classifier.fit(X_train, y_train)
```

▼ GaussianNB  
GaussianNB()

```
[119] from sklearn.metrics import confusion_matrix, accuracy_score
      y_pred = classifier.predict(X_test)
      cm = confusion_matrix(y_test, y_pred)
      cm                                     # Confussion matrix for Gaussian Naive Bayes classification.

      array([[150,  5,  4],
            [ 0, 146,  9],
            [ 10, 27, 128]])
```

```
nb_score = accuracy_score(y_test, y_pred)
print('accuracy', nb_score)                # Model accuracy for Gaussian Naive Bayes.

accuracy 0.8851774530271399
```

## 5.5.2 Random Forest Classification:

### ▼ Random Forest:

```
✓ 2s ▶ from sklearn.ensemble import RandomForestClassifier
forest = RandomForestClassifier(criterion='gini',random_state=0)
forest.fit(X_train, y_train)
```

```
📄 ▼ RandomForestClassifier
RandomForestClassifier(random_state=0)
```

```
[122] y_pred = forest.predict(X_test)
cm=confusion_matrix(y_test,y_pred)
cm                                     # Confussion matrix for Rnadam forest classification.

array([[155,  0,  4],
       [ 0, 154,  1],
       [ 5, 22, 138]])
```

```
[123] forest_score=accuracy_score(y_test,y_pred)
print('accuracy:',forest_score)      # Model accuracy for Random forest.

accuracy: 0.9331941544885177
```

### 5.5.3 Support Vector Machine Classification:

#### ▼ Support Vector Machine:

```
3s ✓ ▶ from sklearn.svm import SVC
kernel=["linear","rbf","poly"]
best_accuracy=0
for k in kernel:
    svc_model = SVC(kernel=k)
    svc_model.fit(X_train, y_train)
    y_pred = svc_model.predict(X_test)
    accuracy=accuracy_score(y_pred,y_test)
    if accuracy>best_accuracy:
        best_accuracy=accuracy
        best_kernel=k
print(f"Best accuracy is {best_accuracy} with best kernel is {best_kernel}")
```

🔗 Best accuracy is 0.9123173277661796 with best kernel is linear

```
0s ✓ [155] svc_model = SVC(kernel="linear")
      svc_model.fit(X_train, y_train)
```

```
▼ SVC
SVC(kernel='linear')
```

```
[157] y_pred = svc_model.predict(X_test)
      svm_score=accuracy_score(y_pred,y_test)
      svm_score                                     # Model accuracy score for Support vector machine

0.9123173277661796
```

```
[158] cm=confusion_matrix(y_test,y_pred)
      cm                                             # Confusion matrix for Support vector machine

array([[155,  0,  4],
       [ 0, 152,  3],
       [12, 23, 130]])
```

### 5.5.4 Accuracy Table:

#### ▼ Accuracy score with different mode:

```
0s model=["Gaussian Naive Bayes", "Random Forest", "SVM"]; accuracy=[]
accuracy=[nb_score, forest_score, svm_score]
pd.DataFrame({"Model": model, "Accuracy": accuracy})
```

	Model	Accuracy
0	Gaussian Naive Bayes	0.885177
1	Random Forest	0.933194
2	SVM	0.912317

According to the above accuracy table, we see that "Random Forest" has higher accuracy than others two models and hence we will take "Random Forest" as our best fitted model and we will use it for new comments.

### 5.5.5 Prediction Function:

```
0s def new_text_prediction(text):
    corpus_copy=[]
    for sentence in final_data['Sentence']:
        corpus_copy.append(sentence)
    # print(corpus_copy[len(corpus_copy)-1])
    # print(len(corpus_copy))
    bag_of_words=corpus_copy
    bag_of_words.append(text)
    corpus_new=bag_of_words
    # print(len(corpus_new))
    # print(corpus_new[len(corpus_new)-1])
    X_new= cv.fit_transform(corpus_new).toarray()
    test_new=X_new[1596]
    test_new=np.reshape(test_new, (-1, 1500)) # where -1 infers the size of the new dimension from the size of the input array.
    # This is the numerical form(array) of the corresponding to new text data.
    # print(test_new)
    y_pred = forest.predict(test_new)
    # print(y_pred)
    if y_pred==[2]:
        print("This is a positive feedback")
    if y_pred==[0]:
        print("This is a negative feedback")
    if y_pred==[1]:
        print("This is a neutral feedback")
```



✓ [162] new\_text\_prediction("okk")  
0s

This is a neutral feedback

✓ [171] new\_text\_prediction("wonderful phone, i have never seen such smartphone like that")  
0s

This is a positive feedback

✓ [164] new\_text\_prediction("nothing new")  
0s

This is a neutral feedback

✓ [165] new\_text\_prediction("bad one")  
0s

This is a negative feedback

## 6 Model Deployment

Model deployment is the process of putting machine learning models into production. This makes the model's predictions available to users, developers or systems, so they can make business decisions based on data, interact with their application (like recognize the insight sentiment of a comment) and so on. Here we used "Flask" to deploy our machine learning classification model so that one can easily classify the new comments.

Flask is a web application framework written in python to build web applications and API architecture

Deploying a machine learning model is a very challenging task. Making the model's predictions available to customers is called deployment. We already built our machine learning model, so there are two steps remaining to deploy our model on web application. These are:

- To save the ML model using Pickle, we pass the model object into the `dump()` function of Pickle. This will serialize the object and convert it into a "byte stream" that we can save as a file called `c2_SentimentAnalysis_Model_pipeline.pkl`. We then store and commit to Git, this model and run it on unseen test data without the need to re-train the model again from scratch.
- **Setting up a Flask web application:** Flask is a web server framework that requires us to organize our code in a specific way. Begin by creating two folders in your MyApp directory called Models and Templates. For the time being, we won't be using the Static folder, but feel free to include any optional CSS code.

You should create a python file on the same root level as the files and name it app.py, just like we did. After that, make an index HTML file and save it in the templates directory. The fact that the app.py and index.html files are empty is unimportant; instead, concentrate on the framework.

We can easily install flask using pip (pip install Flask) and run our web application because flask is just a third-party library. A difficulty arises when the web application is dependent on specific versions of python and other third-party libraries. Imagine that our web application is working great until a third-party library update comes along and changes the names of particular functions. Our programme will become unusable as a result of this. We can use a so-called virtual environment to get around this problem(virtualenv).

Firstly we installed virtualenv on our current Python installation to build a virtual environment.

*!pip install virtualenv*

After installation, we can create a virtual environment. We should generate virtual environment files inside the MyApp directory. Now we change our current directory using CMD or open the MyApp folder and choose New Terminal.

Depending upon your system, use the following commands:

*python3 -m venv virtual (Mac)*

*py -3 -m venv virtual (Windows)*

Now the above command creates a virtual folder with necessary files. For the activation of the environment, use the below command.

*. virtual/bin/activate (Mac)*

*virtualScriptsactivate (Windows)*

Hence we have installed flask and we can see the flask files in our bin folder.

- **Deploy the app on Heroku:**

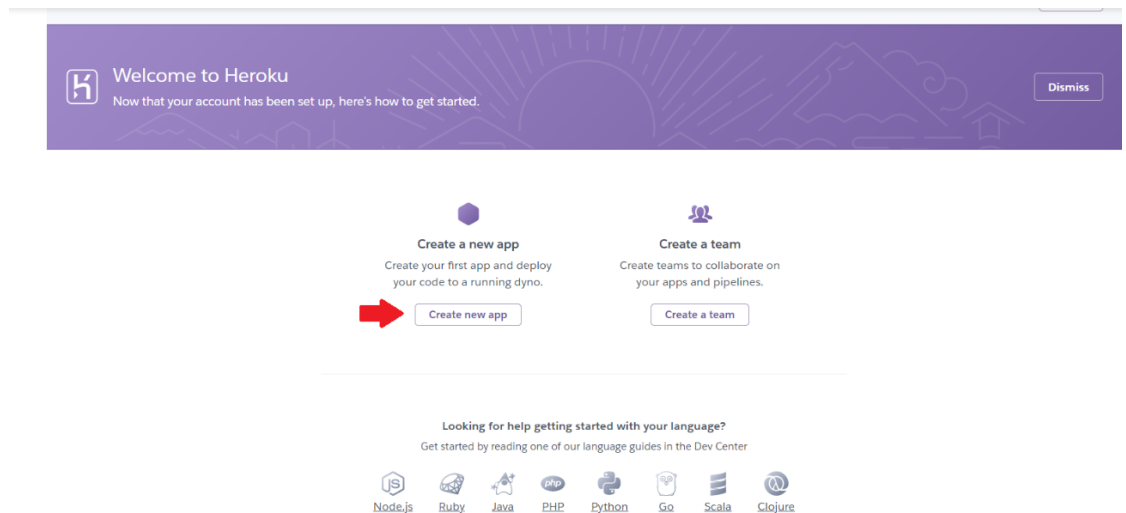
We're ready to start our Heroku deployment now that our model has been trained, the machine learning pipeline has been set up, and the application has been tested locally. There are a few ways to upload your application source code onto Heroku. The easiest way is to link a GitHub repository to your Heroku account.

Although flask is fantastic, they help in local development. It does not handle the kind of queries that a typical web server does. We'll need to install the gunicorn python library to take care of a large number of requests.

We need to tell Heroku to use the gunicorn now that we've installed it. We accomplish this by generating a file called procfile that has no extension (for instance, Procfile.txt is not valid.). The file consists of commands to execute on the startup.

Once files had uploaded to the GitHub repository, we are now ready to start deployment on Heroku. Follow the steps below:

- After sign up on heroku.com then click on Create new app.




– Enter App name and region.


The image shows the "Create New App" form on Heroku. The form has a title "Create New App" at the top. Below the title, there are two main sections: "App name" and "Choose a region". The "App name" section has a text input field with the value "machine-learning-insurance-dev" and a green checkmark icon to the right. Below the input field, there's a message: "machine-learning-insurance-dev is available". The "Choose a region" section has a dropdown menu with the value "United States" and a small icon to the right. Below the dropdown menu, there's a button labeled "Add to pipeline...". At the bottom of the form, there's a large purple button labeled "Create app".


– Connect to your GitHub repository where code is uploaded.

### 3. Connect to your GitHub repository where code is uploaded.

Deployment method

 Heroku Git  
Use Heroku CLI

 GitHub  
Connected

 Container Registry  
Use Heroku CLI

---

App connected to GitHub

Code diffs, manual and auto deploys are available for this app.

Connected to [pavankalyan066/Machine\\_Learning\\_Deployment](#) by [pavankalyan066](#) [Disconnect...](#)

Releases in the [activity feed](#) link to GitHub to view commit diffs

### – Deploy branch

Automatic deploys

Enables a chosen branch to be automatically deployed to this app.

Enable automatic deploys from GitHub

Every push to the branch you specify here will deploy a new version of this app. **Deploys happen automatically:** be sure that this branch is always in a deployable state and any tests have passed before you push. [Learn more.](#)

Choose a branch to deploy

☐ Wait for CI to pass before deploy

Only enable this option if you have a Continuous Integration service configured on your repo.

[Enable Automatic Deploys](#)

---

Manual deploy

Deploy the current state of a branch to this app.

Deploy a GitHub branch

This will deploy the current state of the branch you specify below. [Learn more.](#)

Choose a branch to deploy

[Deploy Branch](#)

### – Wait 5–10 minutes and BOOM

### Manual deploy

Deploy the current state of a branch to this app.

### Deploy a GitHub branch

This will deploy the current state of the branch you specify below. [Learn more.](#)

#### Choose a branch to deploy

 master 

Deploy Branch

Receive code from GitHub

Build **master** 3a3e6e70

Release phase

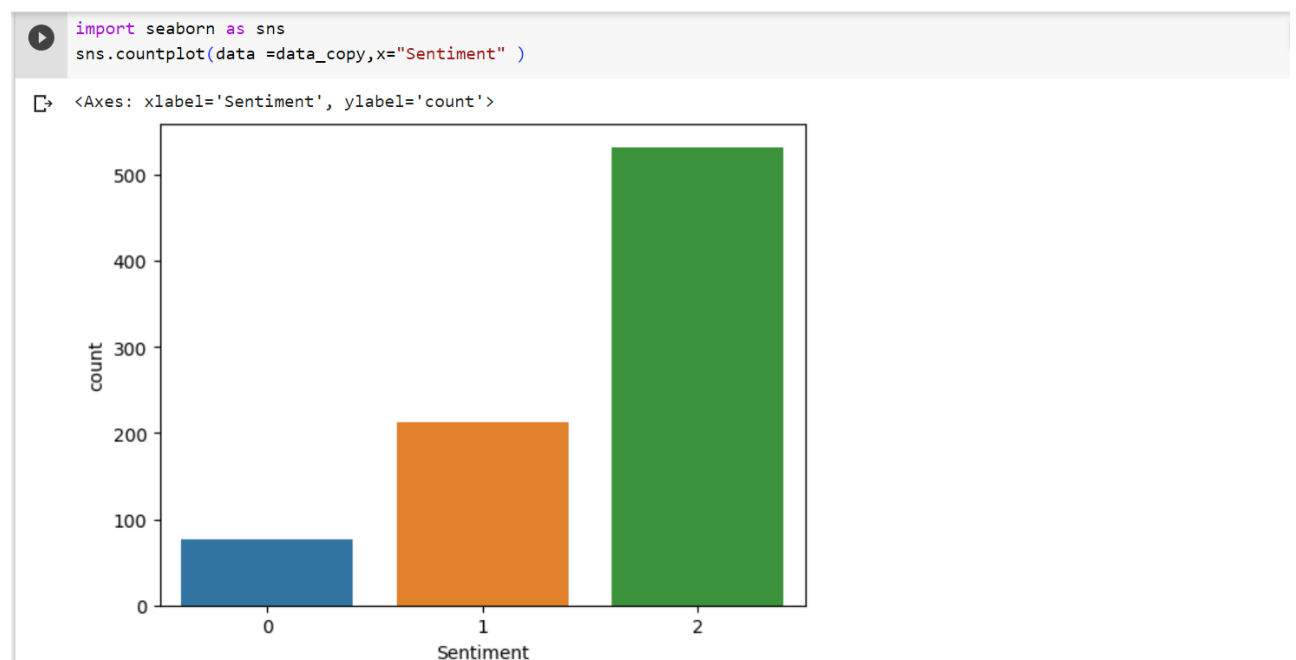
Deploy to Heroku

Your app was successfully deployed.



## 7 Conclusion

YouTube is one of the most popular social media platform where people expressed their thoughts as comments under the corresponding videos. Here we successfully classified the comments on the video "Samsung Galaxy S23 Ultra official trailer" into "positive", "neutral" and "negative" classes. We see that our collected data contains more "positive" sentiment about that smartphone than "negative" sentiment and there are also some "neutral" sentiment which are not going to harm.



It is well that there are majority of "positive" public sentiment and by reading the "negative" sentiment we can suggest modification



to improve the brand image of the product "Sumsung Galaxy S23 Ultra".

Here we listed some negative comments at below:

- **Nothing wen compared to S22 ultra other than 200pxl and processor. No need to upgrade from my S22 ultra.**
- **I have the s22 ultra so I wasn't really impressed until it showed Astrophoto!? The only reason I'm even thinking about upgrading already.**
- **It's basically the same as my S22 Ultra just with slightly small amount of features. So no point of upgrading . . .**
- **If only it had an expandable memory card slot. I would fill the internal storage in no time with 200 megapixel images.**

thereby, we suggest the following modifications to "Sumsung Galaxy S23 Ultra" developing team:

- To include expandable memory card slot for upcoming version.
- To use the better processor for upcoming version.
- To include "Astrophoto" features in the camera.
- To include more features than previous model for upcoming model.

We have also deployed our model on "Heroku" using flask. So when new comments come "Samsung Galaxy S23 Ultra" developers team can classify comments successfully and accordingly make modifications in the upcoming version of "Samsung Galaxy S23 Ultra". The web app link is:

<https://sentiment-analysis-isi-chennai.herokuapp.com/>

## 8 Bibliography

- [1.] Bing Liu. Sentiment Analysis and Opinion Mining, Morgan Claypool Publishers, May 2012.
- [2.] Natural Language Processing with Python by Steven Bird, Ewan Klein, and Edward Loper, First Edition, june 2009.
- [3.] An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, Daniel Jurafsky, James H. Martin, Third Edition draft, Draft of January 7, 2023.