Sentimental Analysis through Speech and text for IMDB Dataset

A Project Report

Submitted in the partial fulfilment of the requirements for the award of the degree of

Bachelor of Technology in

Department of ARTIFICAL INTELLIGENCE AND DATA SCIENCE

By

2000080122 KUMBHA RAJESHBABU

Under the supervision of

Mrs. Lakshmi Lalitha Vuyyuru

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

K L (Deemed to be) University

Green Fields, Vaddeswaram, Guntur District – 522502

(2020-2021)

K L University

DEPARTMENT ARTIFICAL INTELLIGENCE AND DATA SCIENCE

DECLARATION

The project report entitled "**Sentimental Analysis through Speech and text for IMDB Dataset**" is a record of bonafide work done and submitted by **"**Yalavarthi Sikhi (170031425)**"** in partial fulfilment for the award of Bachelor of Technology in Department of Computer Science Engineering to the K L University. The results embodied in this report have not been copied from any other Departments/University/Institute.

KUMBHA RAJESHBABU (2000080122)

K L University

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project report entitled "**Sentimental Analysis through Speech and text for IMDB Dataset**" is a bonafide work done and submitted by "Yalavarthi Sikhi (170031425)" in partial fulfilment for the award of Bachelor of Technology in Department of Computer Science Engineering to the K L University is a record of bonafide work carried out under our guidance and supervision. The results embodied in this report have not been copied from any other Departments /University / Institute.

Signature of the Supervisor

(Assistant Professor)

Signature of the HOD

Mr. V. Hari Kiran

Signature of the External Examiner

ACKNOWLEDGEMENT

The success in this project would not have been possible but for the timely help and guidance rendered by many people. Our sincere thanks to all those who have assisted us in one way or the other for the completion of my project.

Our greatest appreciation to our guide **"Mrs. Lakshmi Lalitha Vuyyuru"** Assistant Professor, Department of Computer Science and Engineering which cannot be expressed in words for her tremendous support, encouragement, and guidance for this project.

We express our gratitude to **Mr. V. Hari Kiran,** Head of the Department for Computer Science and Engineering for providing us with adequate facilities, ways, and means by which we are able to complete this project.

We thank all the members of the teaching and non-teaching staff members, and also who have assisted us directly or indirectly in the successful completion of this project.

ABSTRACT

In the advanced innovative present reality, most of the public is reliant on different surveys for different utilizations and different products. Thus, to examine these audits and comprehend whether they are supporting or refuting about, we can utilize opinion investigation. The days for the remark surveys are currently disappearing and everybody is keen on hearing the audits so as they need not stop their functions. Thus, a slant investigation that does both through brief snippet and text can be used to complete work in sound. Along these lines, in this exploration paper, we are using different Machine Learning and Deep Learning Models like Naive Bayes, Support Vector Machine (SVM), Random Forest, and Multi-Layer Perceptron. Additionally, as of to change over voice to a message we are utilizing a google API and a deep learning technique and exhibit best of two and afterward performing sentiment analysis on those texts and finally obtaining the sentiment (i.e., positive, or negative) of the speech. We are trying to carry out all these processes in real-time.

Keywords: ***Deep Learning, Neural Networks, ReLu, Softmax, SVM, Naïve Bayes, Random Forest, Multi-Layer Perceptron***.

TABLE OF CONTENTS

List of Figures:

List of Tables:

1. INTRODUCTION

In the present electronic world, the scenario had emerged such that all the information or an individual's perspectives are being shared out on public platforms such as social media. The shared or posted information may be text information or an audio clipping or a video (vlog). So, to understand the sense of the information either it is positive or against automatically, the intelligent sentiment analysis procedures have come into the picture. Since the material of the globe keeps on emerging with advanced technologies it is important/ highly prioritized to have a keen observation about all these and the counterparts for tracking should also be updated. Hence, this project had been developed in such a format that it accepts either a texted format input or recognizes the speech in audio format which later will be converted into text format and then proceeds with the normal analysis to segregate it in accordance with its nature.

To personify the usage of product to any user in serene, GUI is developed using Tkinter such that the text can be submitted, or the voice recognition can be started immediately when required and hence the further process can be continued without any discrepancy. The project involves four different algorithms namely Naïve Bayes algorithm, Support Vector Machine (SVM) algorithm, Random Forest algorithm, and Multi-Layer Perceptron algorithm to examine the input and segregate them too positive or negative. And at the end we calculate the best out of 4 algorithms.

Algorithm	Definition	Formula
SVM	The objective of the support vector machine algorithm is to seek out a hyperplane in an N-dimensional space (N — the number of features) that distinctly classifies the info points.	K (x, xi) = sum (x ∗xi)
Naive Bayes	Naive Bayes classifiers are a set of classification algorithms supported Bayes' Theorem. It is not one algorithm but a family of algorithms where all of them share a standard principle, i.e., every pair of features being classified is independent of every other.	P(A|B) =P(B|A)P(A) / P(B)
Random Forest	Random forest may be a supervised learning algorithm which is employed for both classifications also as regression. But however, it is mainly used for classification problems. As we all know that a forest is formed from trees and more trees means more robust forest.	MSE = $\frac{1}{N}\sum_{i=1}^{N}(fi - yi)^2$
Multi-Layer Perceptron	A multilayer perceptron (MLP) might be a class of feed forward fake neural organization (ANN). MLP uses a directed learning strategy called backpropagation for preparing. Its numerous layers and non-straight actuation recognize MLP from a direct perceptron.	y = f(WxT + b).

Table 1: Brief on Algorithms

2. LITERATURE SURVEY

The process of Sentiment Analysis involves the construction of the input vector space from the existing document vector space. Mainly there are two approaches to carry out vector space mapping. The machine learning based, or statistical based feature extraction methods are widely used because extraction of features is done by applying statistical measures directly. Earlier works on sentiment classification using machine learning approaches were carried by Pang et al. in 2002 [1].

Similar work was done by Tripathi et al. [2], where TF, TF-IDF was used for the conversion of the text file to a numerical vector. Experimentation was done with n-gram approaches and its combination are tried to get the best results.

Identifying the semantics or the meaning of the text by a machine learning algorithm is a challenging task. Lexicon features are used in this regard to extract the opinions expressed in the text. Sarcasm detection is one of the major advantages of choosing lexicon features. Anukarsh et al. [3]

Melville et al. [4], worked on extracting features using lexicon methods. Positive and negative word counts that are present in the text were used as the background lexicon knowledge and then the probability that a document belongs to a particular class was calculated.

With the social media platforms such as Twitter, Facebook, Instagram, and WhatsApp taking the communication world by storm, it has become imperative that the data residing across these social media platforms will convey insightful information about the opinion, mood, and sentiment of the people over any product, idea, or policies. Several works have been performed earlier to analyse the twitter contents and perform opinion mining over twitter data. The authors of [5] have proposed an approach that uses deep convolution neural network to analyse the twitter feed.

The usage of twitter has kindled more research work towards understanding the sentiments using twitter data. One such work discussed in [6] uses a hybrid framework that uses a genetic algorithm-based approach to perform sentiment analysis.

C.D. Santos et al. [7] had proposed a replacement deep convolutional neural network that exploits from character to sentence-level information to perform sentiment analysis of short texts. Sentiment analysis of short texts like single sentences and Twitter messages is challenging due to the limited contextual information that they normally contain. Effectively solving this task requires strategies that combine the tiny text content with prior knowledge and use quite just bag-of-words.

A. Ortigosa et al. [8] presented a replacement method for sentiment analysis in Facebook that, ranging from messages written by users, supports: (i) to extract information about the users' sentiment polarity (positive, neutral, or negative), (ii) to identify major emotional changes by modelling the users' normal sentiment polarity. They have implemented this method in Sent Buk, a Facebook application

R. Ahmad et al. [9] proposed a new technique Aspect based sentiment analysis (ABSA) is a fine-grained text analysis technique that extracts individuals and aspects from text and analyses the sentiments expressed against them. Previous additions to this segment have mainly concentrated on data from brief product, restaurant, and service reviews.

H. M. Kumar et al. [10] developed an algorithm Sentiment analysis or opinion mining is an automatic process to acknowledge opinion, moods, emotions, attitude of people or communities through tongue processing, text analysis, and linguistics.

E. Cambria et al. [11] has analysed that the SenticNet is a freely accessible semantic and affective resource for emotion analysis at the definition level. SenticNet 3 uses ‘energy flows' to link different sections of extended common and common-sense information representations to one another, rather than graph-mining and dimensionality-reduction techniques.

M. E. Moussa et al. [12] has surveyed because of the increased research interest in information engineering tasks such as Opinion Summarization, the need for efficient processing of this vast amount of data has evolved. This poll summarizes current public sentiment.

I. Hemalatha et al. [13] mainly discussed about the user's reviews represent a useful approach to sentiment analysis. The focus of this research paper is on the pre-processing techniques used in conjunction with specially developed algorithms to perform sentiment analysis.
A. S. Manek et al. [14] proposed for sentiment classification of a broad movie review data set, a Gini Index-based feature selection method with Support Vector Machine (SVM) classifier is proposed. The results show that our Gini Index system performs better in terms of classification accuracy and error rate.

A Kennedy et al. [15] developed approaches for assessing the emotion conveyed in a movie review are provided. An analysis may have a positive, negative, or neutral semantic orientation. We investigate the effects of valence shifters on the classification of feedback. Negations, intensifiers, and diminishers are the three forms of valence shifters studied using SVM and other Machine Learning algorithms.

Machine Learning: Machine Learning is usage of man-made brainpower (AI) that enables systems to thus take in and improve without being expressly customized. AI focuses works for the improvement of PC programs that can get to data and use it to adapt expressly. For example, when gathering pictures, the machine is set up with various arrangements of pictures and their relating marks, where the image is the information, and its correct name is the yield.

Artificial Neural Network: Artificial Neural Networks are the most popular machine learning algorithms today. The invention of those Neural Networks happened within the 1970s, but they need achieved huge popularity thanks to the recent increase in computation power due to which they are now virtually everywhere. In every application that you simply use, Neural Networks power the intelligent interface that keeps you engaged.

Speech Recognition:

Speech recognition is the ability of a device or system to recognize spoken words and convert them into readable text. Speech recognition is utilized in different fields of research in computer science, linguistics, Natural language processing, and computer engineering. Many modern devices are associated with speech recognition functions for enabling hands-free use of a device.

Existing Models for Speech Recognition:

1. Speech Recognition Using Google Cloud Speech API.
2. Speech Recognition Using Deep Neural Networks.
3. Speech Recognition Using Hidden Markov Models.

These existing models can be utilized as an application of speech to text translation for further identification of objects.

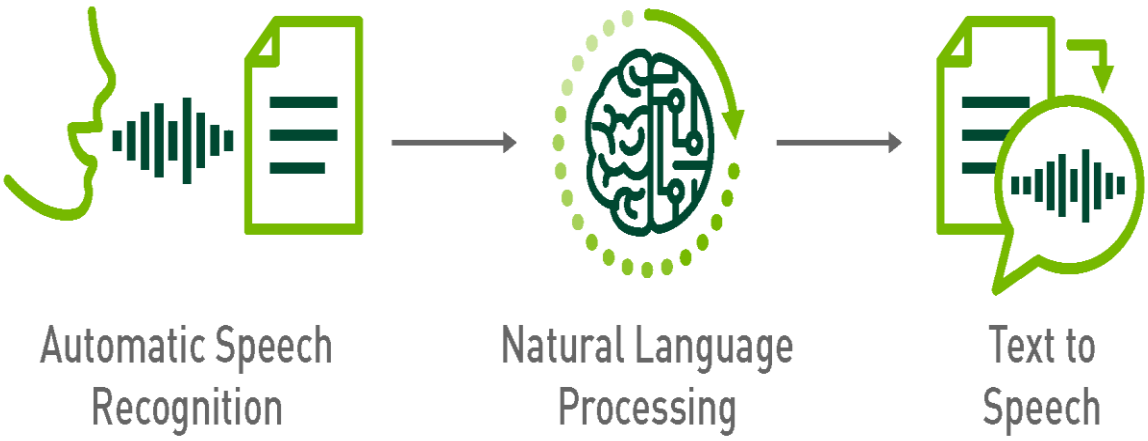


Fig.1.Workflow of Speech Recognition.

3. PROPOSED WORK

3.1 Importing all the packages:

To run this model, you need to import all the essential packages and libraries. To obtain the speech input, the user needs to import the Speech Recognition package which contains many inbuilt methods like the listen method and the recognizer google method. The recognizer method helps to recognize the speech and convert it into the desired text.

Later, you need to install the PyAudio package, this is used to record the voice data of the user through the microphone of the device. When you run the Speech Recognition module it takes the input from speech and it is converted to text and will detect Sentimental Analysis and as well as it can take input in text and can detect the Sentimental Analysis.

Module 1: Developing a Speech Recognition Model.

Module 2: Developing an Sentimental Analysis Model.

Module 3: Integrating Speech Recognition and Sentimental Analysis Model.

Requirements:

1. Python (version 3.7 or higher).
2. Google Colab.
3. Jupyter notebook.
4. SpeechRecognition package.
5. Pyaudio package.
6. LabelImg Software.
7. tkinter

3.2 Methodology:

The algorithms used for the classification task and the methods column lists the basic mechanisms for feature engineering. We obtained acceptable results with common Machine Learning and Deep Learning algorithms such Naive Bayes (NB), decision tree classifier (DT), and support vector machines (SVM), Multi-Layer Perceptron. However, after analysing the state of-the-art models on this dataset, we decided to go further and attempt to replicate those results, using only the models with classical machine learning algorithms. We found that Multi-Layer Perceptron has performed well on this dataset.

3.2.1 SUPPORT VECTOR MACHINE (SVM)

Support Vector Machine (SVM) is an extremely good, supervised learning algorithm. SVM algorithm creates decision boundaries that can separate n-dimensional space into different classes so that new points can be placed in the correct classism is used in real life for text, image classification, face detection etc.

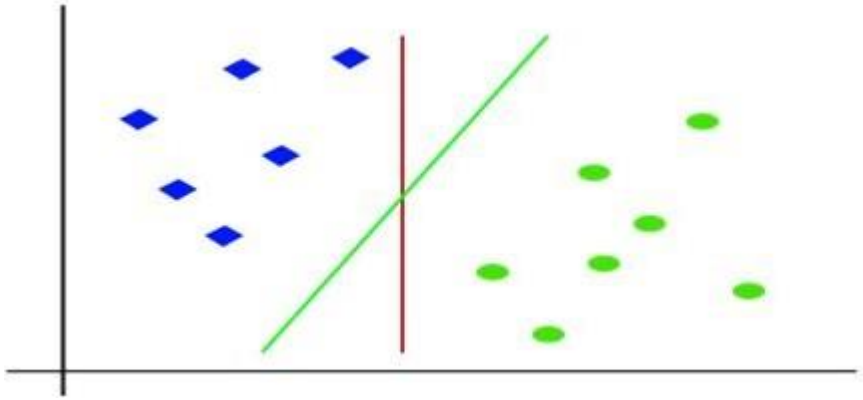
Support Vector Machine is a supervised machine learning algorithm capable ofperforming classification. SVM is of two types:

1)Linear SVM

2)Non-Linear SVM

Linear SVM

The linear Support Vector Machine classifier works by drawing a line between the two classes.



Dataset divides into classes in Linear SVM

There are some important concepts to know in the SVM algorithm to divide the dataset into classes. They are Support Vectors, Hyperplane, Margin.

Hyperplane

The SVM model is fundamentally a portrayal of various classes in a hyperplane in multidimensional space. The hyperplane will be created in an iterative way by SVM so the error can be limited. SVM or Support vector machine is the classifier that supports the edge. The target of a classifier in our model underneath is to find a line or (n-1) estimation hyper-plane that disconnects the two classes present in the n-dimensional space.The objective of SVM is to separate the datasets into classes to locate a most maximum marginal hyperplane (MMH). The hyperplane which has maximum margin is called the optimal hyperplane.

Support Vectors

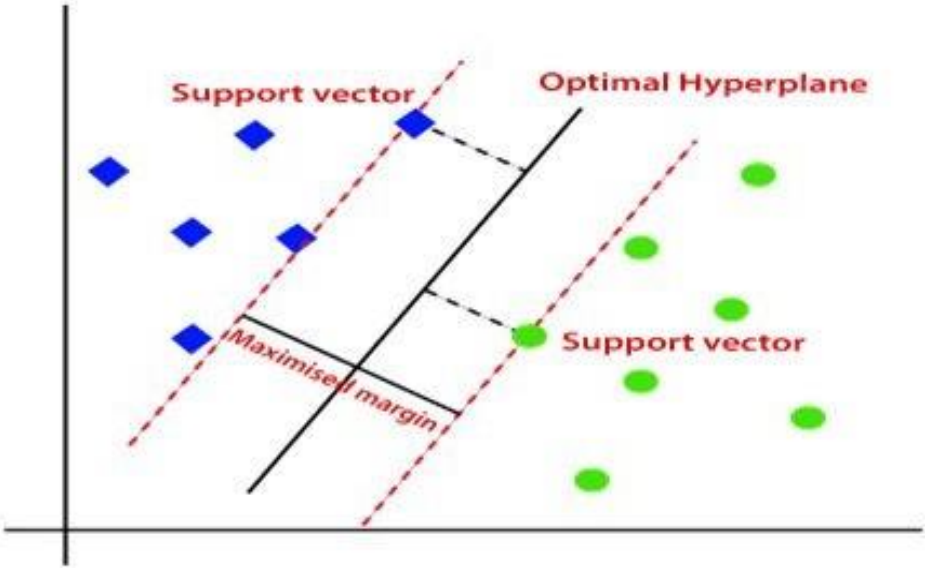
Data Points that are nearest to the hyperplane are called support vectors. Lines separating

them will be defined with the help of data points. With the help of datapoints, separating lines will be defined.

Margin

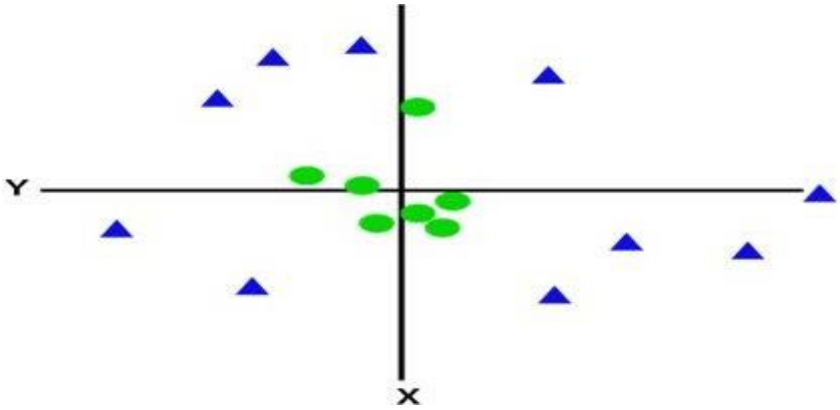
The gap between two lines or vectors on the closet data points of different classes is called Margin. which can be said as the perpendicular distance from the line to the support vectors. Large margin is said to be a good margin and a small margin is a bad margin.

- Support vector Machine will generate hyperplanes iteratively that segregates the instructions withinside the quality way.
- Then, it will choose the hyperplane that separates the classes correctly.



Non-Linear SVM

For Non-linear data, we cannot draw the straight line for data points of the non-inear svm



To separate these dimensions we need to add one more dimension, for linear data we use two dimensions for non-linear data, we use three dimensions. So the SVM divides the data points in the following way

Advantages of SVM:

- SVM classifiers can give good accuracy even in high dimensional space.
- SVM classifiers use very less memory because it uses a small part of training points. uses very less memory.
- SVM is effective when the number of dimensions is greater than the number of samples.
- SVM works properly with a clean margin of separation.

Disadvantages of SVM:

- SVM classifiers do not work well with overlapping classes.
- SVM does not carry out nicely while we take huge datasets due to the fact the training time is high.

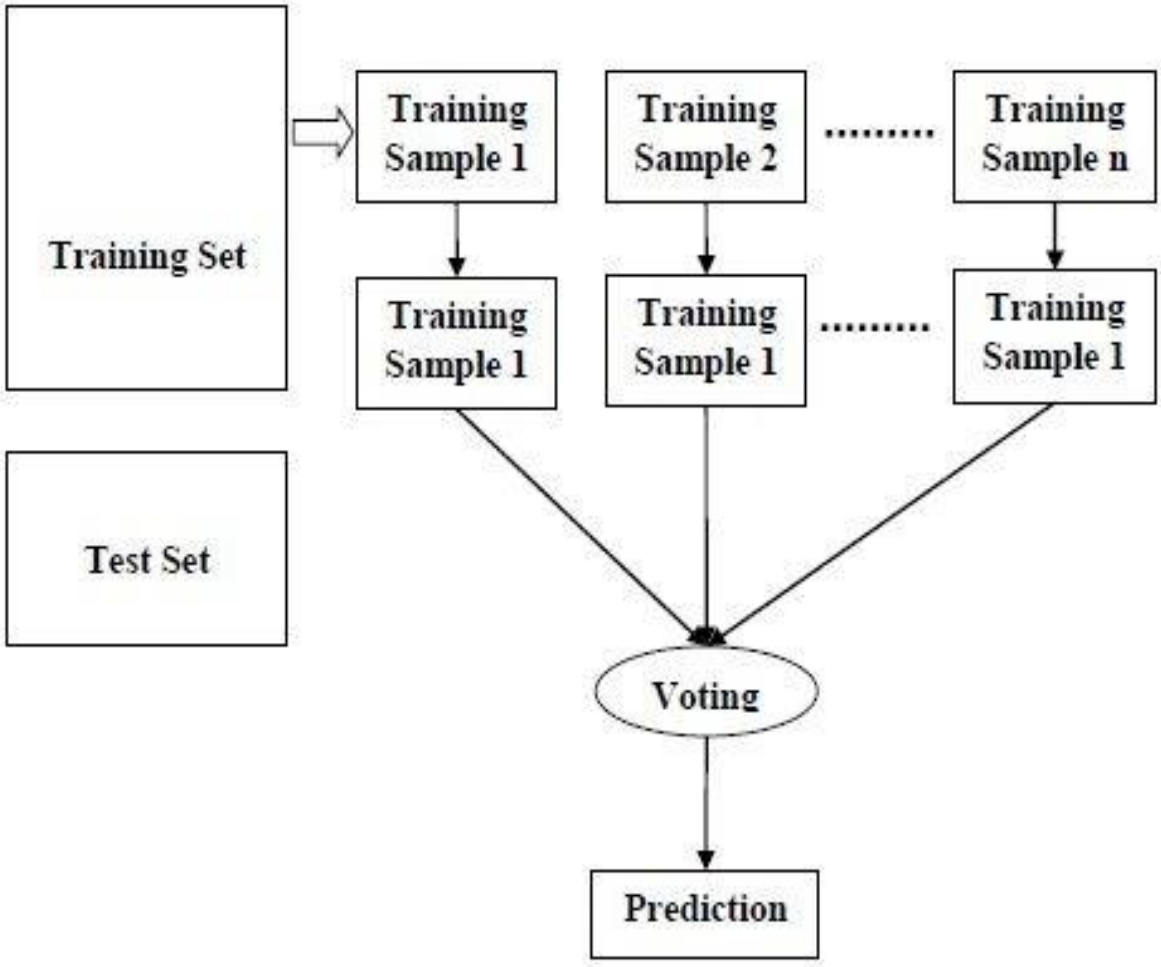
3.2.2 RANDOM FOREST

Random Forest is a supervised learning technique that can be used for Classification problems in Machine Learning. It is built based on a concept called ensemble learning, where multiple classifiers are combined to solve a complex problem and improve amodel's performance.

Random Forest is a classifier which contains many decision trees built on a given dataset and it takes average of output to improve accuracy. It uses an ensemble method which. gives better results than a single decision tree because over-fitting is reduced by averaging the figure.

The number of trees is proportional to accuracy and also prevents the problem of overfitting.

The following diagram describes about the working of Random Forest.



Algorithm for Random Forest

Random Forest works in two phases, one is to create a random forest by predicting the Ndecision trees and the second is to predict each decision tree created in the first phase.

Step-1: Selecting k data points randomly from the training set. Step-2: Creating the decision trees using the selected data points.Step-3: Choose the N decision trees to create for the model.

Step-4: Repeat step-1& step-2

Step-5: For new data points, find the predictions of each decision tree, and assign the newdata points to the category that wins the majority votes.

The random forest algorithm combines multiple trees to predict which class the output belongs to. Some decision trees might predict correct output while some may not. But all together, all the trees predict the right output. Therefore, below are two assumptions for a Random forest classifier:

- In feature variables, actual values should be present to get better accuracy.
- The predictions from each tree must have very low relations.

Advantages of Random Forest

- Random Forest overcomes the problem of overfitting by averaging or combiningthe results of different decision trees.
- Random Forest gives high accuracy and is very flexible.
- Random Forest has less variance than one decision tree.
- Random Forest works well for large datasets than one single decision tree
- Data scaling is not required in Random Forest.
- Random Forest maintains high accuracy when data is missing.

Disadvantages of Random Forest

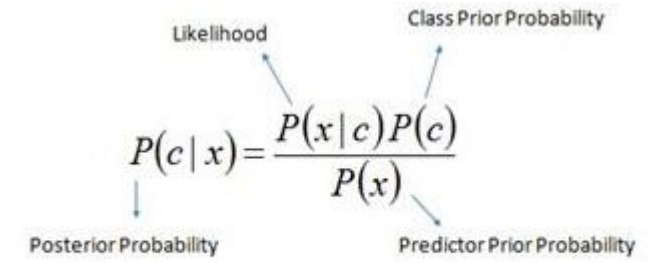
- Random Forests are not more suitable for regression tasks.
- Complexity is more for Random Forest because to create decision trees is muchharder for the algorithm and time complexity is also more.
- More Computational resources are required for the Random Forest Algorithm.
- Prediction is more difficult for Random Forest and very time consuming whencompared to other algorithms.

3.3.3 NAIVE BAYES

Naive Bayes is a supervised learning algorithm, which depends on Bayes hypothesis. In real life it is used in text classification. Naïve Bayes Classifier is one of the basic and best Classification calculations which helps in building the quick AI models that can make accurate predictions. It is based on the probability of an event occurring. So, it is called a

probabilistic classifier. Real life examples of Naïve Bayes Algorithms are Sentimental analysis, spam filtration, classification of articles etc. It is called Naïve on the grounds that it expects that the event of a specific component is independent of the event of different features.

Naive Bayes is a machine learning model dependent on the Bayes hypothesis. It is a straightforward classification method; however it has high usefulness. It is useful when the dataset is large. Bayes hypothesis gives a method of computing the likelihood, P(c|x), from P(c), P(x), and P(x|c). Bayes classifiers accept that the impact of the estimation of an indicator (x) on a given class (c) is free of the estimations of different features.

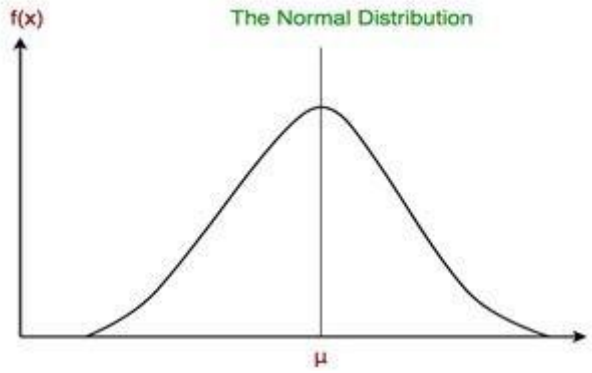


$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

P(c|x) is the likelihood of class (target) given indicator (property).

P(c) is the earlier likelihood of class.

P(x|c) is the probability which is the likelihood of the indicator given class.P(x) is the earlier likelihood of indicator.



Bayes' Theorem finds the likelihood of an occasion happening given the likelihood of another occasion that has just happened. It is used for calculating conditional probabilities. Bayes hypothesis is expressed numerically as the accompanying condition: Naïve bayes has three types of models they are Gaussian, Bernoulli, Multinomial.

Gaussian:

The Gaussian model follows the features of normal distribution. In this model, predictors take continuous values instead of discrete values. At that point the model accepts thatthese values are sampled from the Gaussian distribution.

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

Bernoulli:

The Bernoulli classifier is like the Multinomial classifier, however the predictor variables are the independent Booleans variables. This model can be utilized for classification tasks.

Multinomial:

The Multinomial Naive Bayes classifier is utilized when the information is multinomial distributed. It is fundamentally utilized for classification. The classifier utilizes the frequency of words for the predictors.

Advantages of Naive Bayes

- Naive Bayes is easy and fast to predict the class of test data set. It also performswell in the multi class prediction.
- It can be used for Binary class predictions as well as multi class

predictions and itperforms well for multiclass predictions as compared to other algorithms.

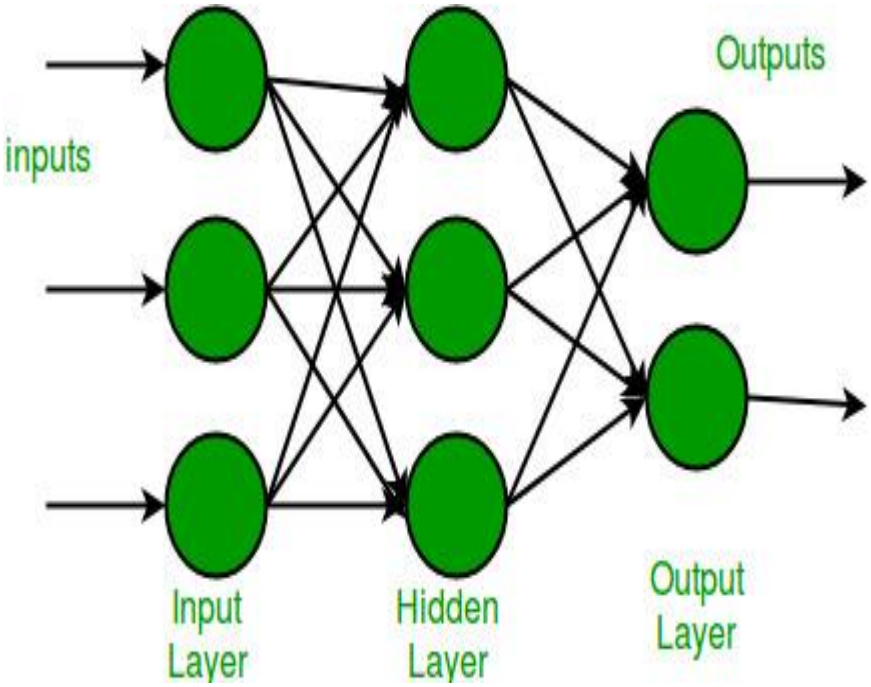
- This algorithm is mainly used for text classification problems.

Disadvantages of Naive Bayes

- Naive Bayes algorithm feels that every feature in the data set is independent andunrelated, so it doesn't learn relationships between the features.
- Naive Bayes is also known as a bad estimator because it doesn't take theprobability of the predictors too seriously.

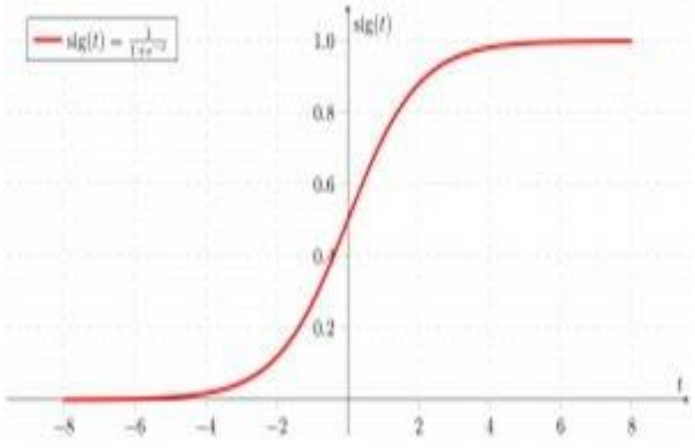
3.3.4 Multi-Layer Perceptron

A Multi-Layer Perceptron (MLP) or Multi-Layer Neural Network contains one or additional hidden layers (apart from one input and one output layer). whereas one layer perceptron will solely learn linear functions, a multi-layer perceptron also can learn non – linear functions.

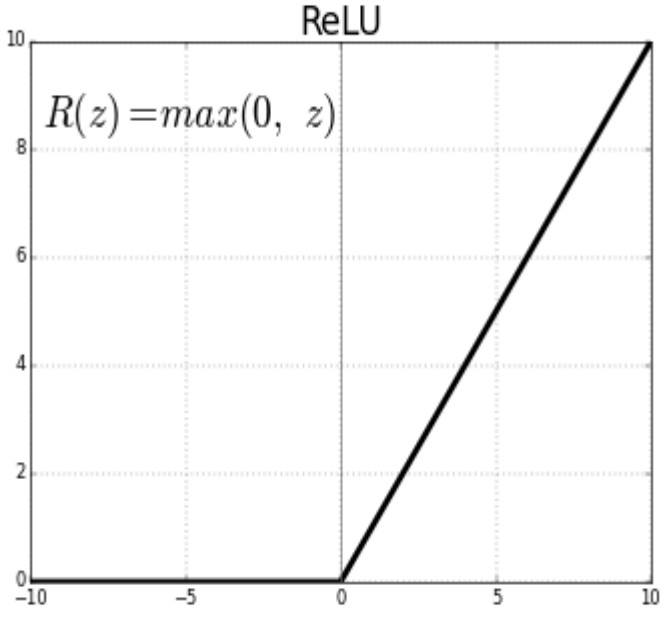


This nerve cell takes as input x1,x2,….,x3 (and a +1 bias term), and outputs f(summed inputs +bias), where f(.) known as the activation perform. the most perform of Bias is to supply each node with a trainable constant price (in addition to the conventional inputs that the node receives). each activation perform (or non-linearity) takes one range and performs a particular fastened computing thereon. There is area unit many activation functions you will encounter in practice:

Sigmoid: takes real-valued input and squashes it to vary between zero and one.



ReLu: ReLu stands for corrected Linear Units. It takes real-valued input and thresholds it to zero (replaces negative values to zero).



Dataset and Setup

The Dataset contains the crude content of 50,000 IMDb movie reviews, each marked as positive or negative, with no different highlights. The test set contains the content of 50,000 unlabelled movie reviews. For assessment and model determination. For pre-preparing, we cleaned the information by eliminating HTML labels, non-word characters, accents, and stop words. At that point we had done tokenizing. For specific techniques, we saw that restricting the quantity of words helped the order task. We accepted this is on the grounds that a portion of the top words, for example, zzz (73286 events), did not give semantic substance to the investigation of suppositions. We found that restricting the words inside a specific scope of recurrence assisted with eliminating clamor and improve our scores. Thus, we eliminated the 1 to 5 least incessant words and simply permitted the 4000 to 6000 most continuous words. Additionally, the quantity of events of a word does not really suggest its significance all through the reports. Accordingly, we utilized TF-IDF to evaluate this worth. Common data was additionally used to choose the highlights with more data concerning the objective variable. At last, the pre-handled information is utilized in certain models.

Review	Sentiment
A rating of "1" does not begin to express how dull, depressing and relentlessly bad this movie is.	negative
If you like original gut-wrenching laughter you will like this movie. If you are young or old then you will love this movie, hell even my mom liked it. Great Camp!!!	positive
This is a great film. Touching and strong. The direction is without question breathless. Good work to the team. I feel so sorry for Marlene, By the grace of God go you or I	positive
Hated it with all my being. Worst movie ever. Mentally- scarred. Help me. It was that bad. TRUST ME!!!	Negative

Table 2: Sample data from dataset

4. BLOCK DIAGRAM

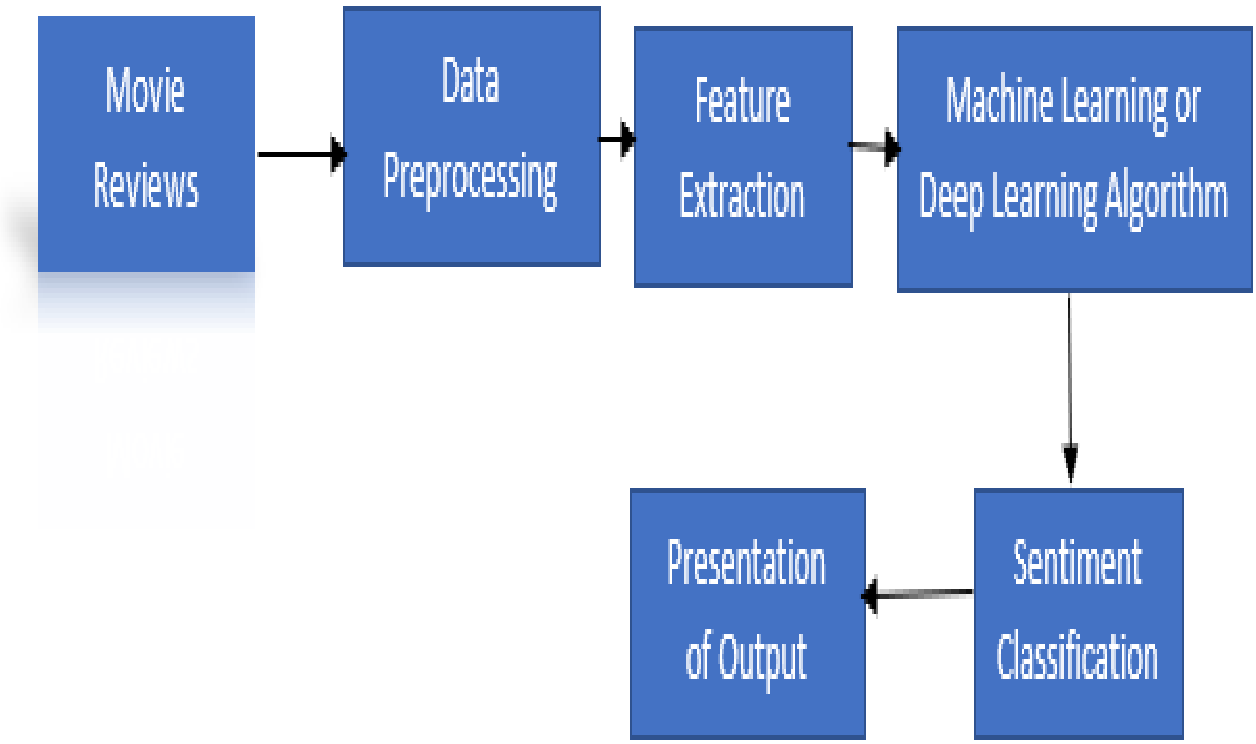


Fig.2. Block Diagram of the process

5. IMPLEMENTATION

Heart Attack Prediction

```
import numpy as np

import pandas as pd

df=pd.read_csv('heart.csv')

df

df.head(5)

import numpy as np

import pandas as pd

import %matplotlib pyp as plt

import seaborn as sns
```

Sentimental Analysis using Random Forest

```
import numpy as np

import pandas as pd

import %matplotlib pyp as plt

import seaborn as sns

from sklearn.linear_model import LogisticRegression

clf = LogisticRegression()
```

```
    clf.fit(X_train, y_train)
    from sklearn.metrics import accuracy_score
    Y_pred = clf.predict(X_test)
    acc=accuracy_score(y_test, Y_pred)
    print('Accuracy is',round(acc,2)*100,'%')
    Sentimental Analysis using Random Forest
X = df.drop('output', axis=1)
Y = df['output']
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.2,random_state=42)
X_train.shape
```

Sentimental Analysis using Logistic Regression

```
from sklearn.linear_model import LogisticRegression
clf = LogisticRegression()
clf.fit(X_train, y_train)
from sklearn.metrics import accuracy_score
Y_pred = clf.predict(X_test)
acc=accuracy_score(y_test, Y_pred)
print('Accuracy is',round(acc,2)*100,'%')
minAge=min(df.age)
maxAge=max(df.age)
```

```
meanAge=df.age.mean()

print('Min Age :',minAge)

print('Max Age :',maxAge)

print('Mean Age :',meanAge)

import numpy as np

import pandas as pd

import seaborn as sns

Young = df[(df.age>=29)&(df.age<40)]

Middle = df[(df.age>=40)&(df.age<55)]

Elder = df[(df.age>55)]

plt.figure(figsize=(23,10))

sns.set_context('notebook',font_scale = 1.5)

sns.barplot(x=['young ages','middle ages','elderly
ages'],y=[len(Young),len(Middle),len(Elder)])

plt.tight_layout()

colors = ['red','green','yellow']

explode = [0,0,0.1]

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

sns.set_context('notebook',font_scale = 1.2)

plt.pie([len(Young),len(Middle),len(Elder)],labels=['young ages','middle ages','elderly
ages'],explode=explode,colors=colors, autopct='%1.1f%%')

plt.tight_layout()
```

8. RESULTS AND ANALYSIS

The Dataset is prepared with 80% training data while 20% of validation data. Different models are tested. Using the general settings described above, we reported the top models with each algorithm in Figure 2. There was not a major difference in runtime. In general, RF and SVM did not perform well on this dataset. Naïve Bayes performed significantly better. And Multi-Layer Perceptron has performed better than other algorithms. We found that Multi-Layer Perceptron performed best on this dataset, providing an accuracy of 88%, respectively. The accuracy of the above-mentioned algorithms is obtained by testing the movie reviews which were given in IMDB Dataset. The sample reviews are “A rating of "1" doesn't begin to precise how dull, depressing and relentlessly bad this movie is.” and “This may be a great film. Touching and powerful. The direction is without question breathless. Good work to the team. I feel so pitying Marlene, By the grace of God go you or I”.

In [1]:
```
import numpy as np
import pandas as pd
df=pd.read_csv('heart.csv')
df
```

Out[1]:

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	..
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	0

303 rows × 14 columns

In [2]:
```
df.head(5)
```

Out[2]:

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       303 non-null    int64
 1   sex       303 non-null    int64
 2   cp        303 non-null    int64
 3   trtbps    303 non-null    int64
 4   chol      303 non-null    int64
 5   fbs       303 non-null    int64
 6   restecg   303 non-null    int64
 7   thalachh  303 non-null    int64
 8   exng      303 non-null    int64
 9   oldpeak   303 non-null    float64
 10  slp       303 non-null    int64
 11  caa       303 non-null    int64
 12  thall     303 non-null    int64
 13  output    303 non-null    int64
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
```

```
df.describe()
```

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000

```
df.isnull()
```

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
0	False	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	False	False	False	False	False	False	False	False	False	False	False	False	False	False
299	False	False	False	False	False	False	False	False	False	False	False	False	False	False
300	False	False	False	False	False	False	False	False	False	False	False	False	False	False
301	False	False	False	False	False	False	False	False	False	False	False	False	False	False
302	False	False	False	False	False	False	False	False	False	False	False	False	False	False

303 rows × 14 columns

Fig 4: Output of Logistic Regression

```
In [21]: from sklearn.linear_model import LogisticRegression
         clf = LogisticRegression()
         clf.fit(X_train, y_train)
         from sklearn.metrics import accuracy_score
         Y_pred = clf.predict(X_test)
         acc=accuracy_score(y_test, Y_pred)
         print('Accuracy is',round(acc,2)*100,'%')

         Accuracy is 89.0 %

         C:\Users\rajes\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:763
         (status=1):
         STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

         Increase the number of iterations (max_iter) or scale the data as shown in:
             https://scikit-learn.org/stable/modules/preprocessing.html
         Please also refer to the documentation for alternative solver options:
             https://scikit-learn.org/stable/modules/linear_model.html#logistic-regressio
           n_iter_i = _check_optimize_result(
```

Fig 5: Output of Random Forest

```
X = df.drop('output', axis=1)
Y = df['output']
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.2,random_state=42)
X_train.shape
```

```
(242, 13)
```

```
import numpy as np
import pandas as pd
df=pd.read_csv('heart.csv')
df.head(50)
```

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
5	57	1	0	140	192	0	1	148	0	0.4	1	0	1	1
6	56	0	1	140	294	0	0	153	0	1.3	1	0	2	1
7	44	1	1	120	263	0	1	173	0	0.0	2	0	3	1
8	52	1	2	172	199	1	1	162	0	0.5	2	0	3	1
9	57	1	2	150	168	0	1	174	0	1.6	2	0	2	1
10	54	1	0	140	239	0	1	160	0	1.2	2	0	2	1
11	48	0	2	130	275	0	1	139	0	0.2	2	0	2	1
12	49	1	1	130	266	0	1	171	0	0.6	2	0	2	1
13	64	1	3	110	211	0	0	144	1	1.8	1	0	2	1
14	58	0	3	150	283	1	0	162	0	1.0	2	0	2	1
15	50	0	2	120	219	0	1	158	0	1.6	1	0	2	1

Fig 6: Output of Naive Bayes

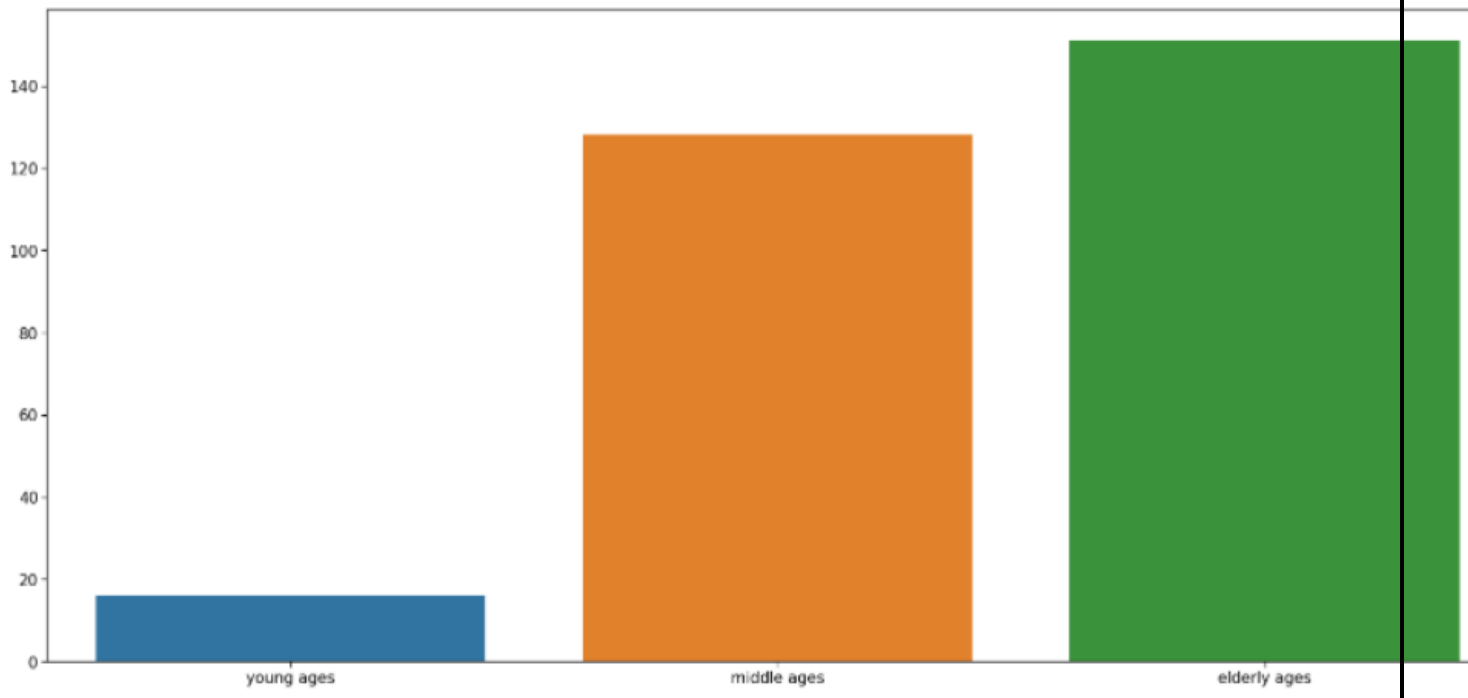
```
In [27]: minAge=min(df.age)
         maxAge=max(df.age)
         meanAge=df.age.mean()
         print('Min Age :',minAge)
         print('Max Age :',maxAge)
         print('Mean Age :',meanAge)

         Min Age : 29
         Max Age : 77
         Mean Age : 54.366336633663366
```

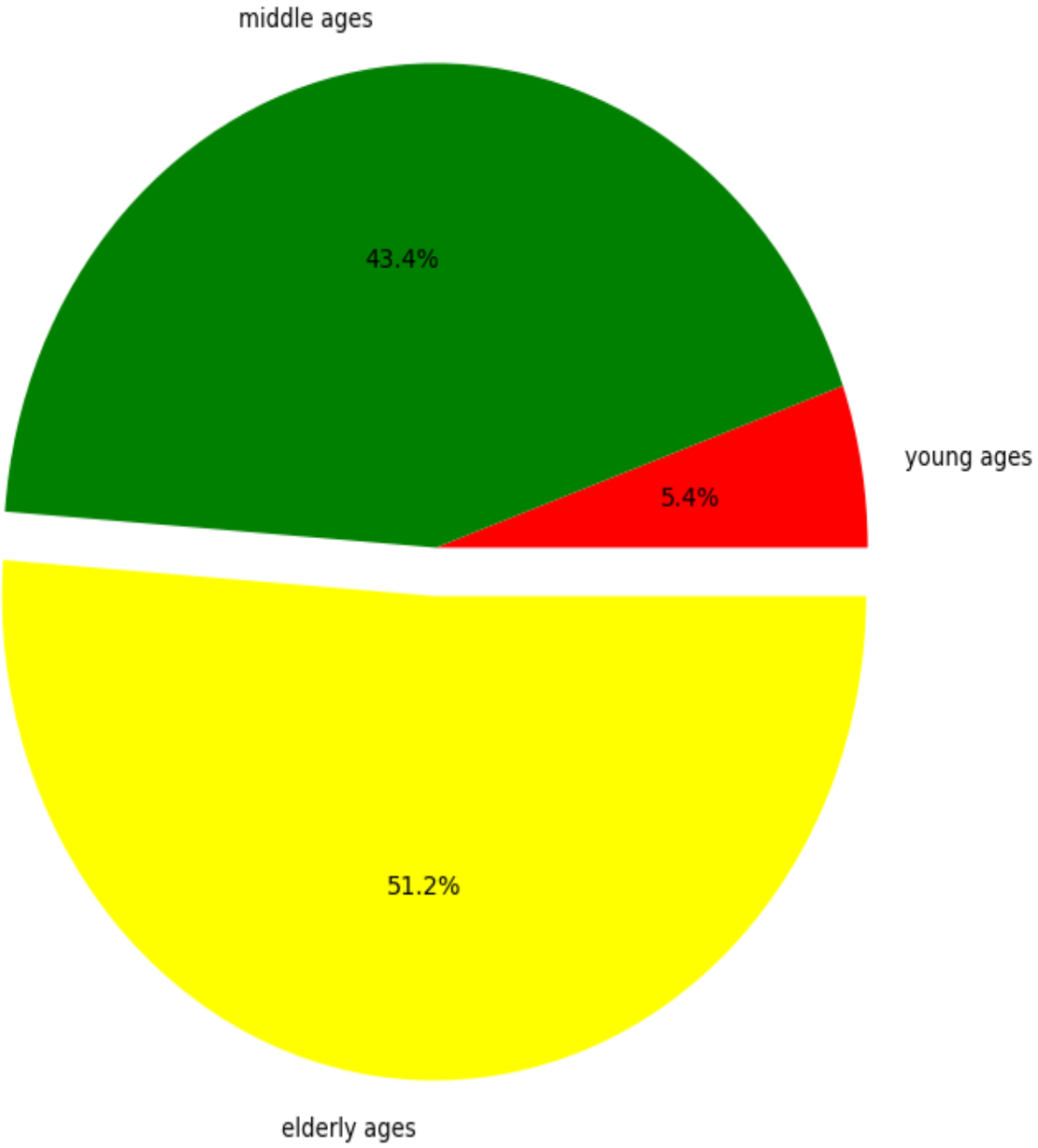
Fig 7: Accuracies graph

During the training phase, the model is trained using the IMDB Dataset. You will get different accuracy rates based on the number of iterations that you perform on the model. During every iteration, the model will compare the specified Review.

```
import numpy as np
import pandas as pd
import seaborn as sns
Young = df[(df.age>=29)&(df.age<40)]
Middle = df[(df.age>=40)&(df.age<55)]
Elder = df[(df.age>55)]
plt.figure(figsize=(23,10))
sns.set_context('notebook',font_scale = 1.5)
sns.barplot(x=['young ages','middle ages','elderly ages'],y=[len(Young),len(Middle),len(Elder)])
plt.tight_layout()
```

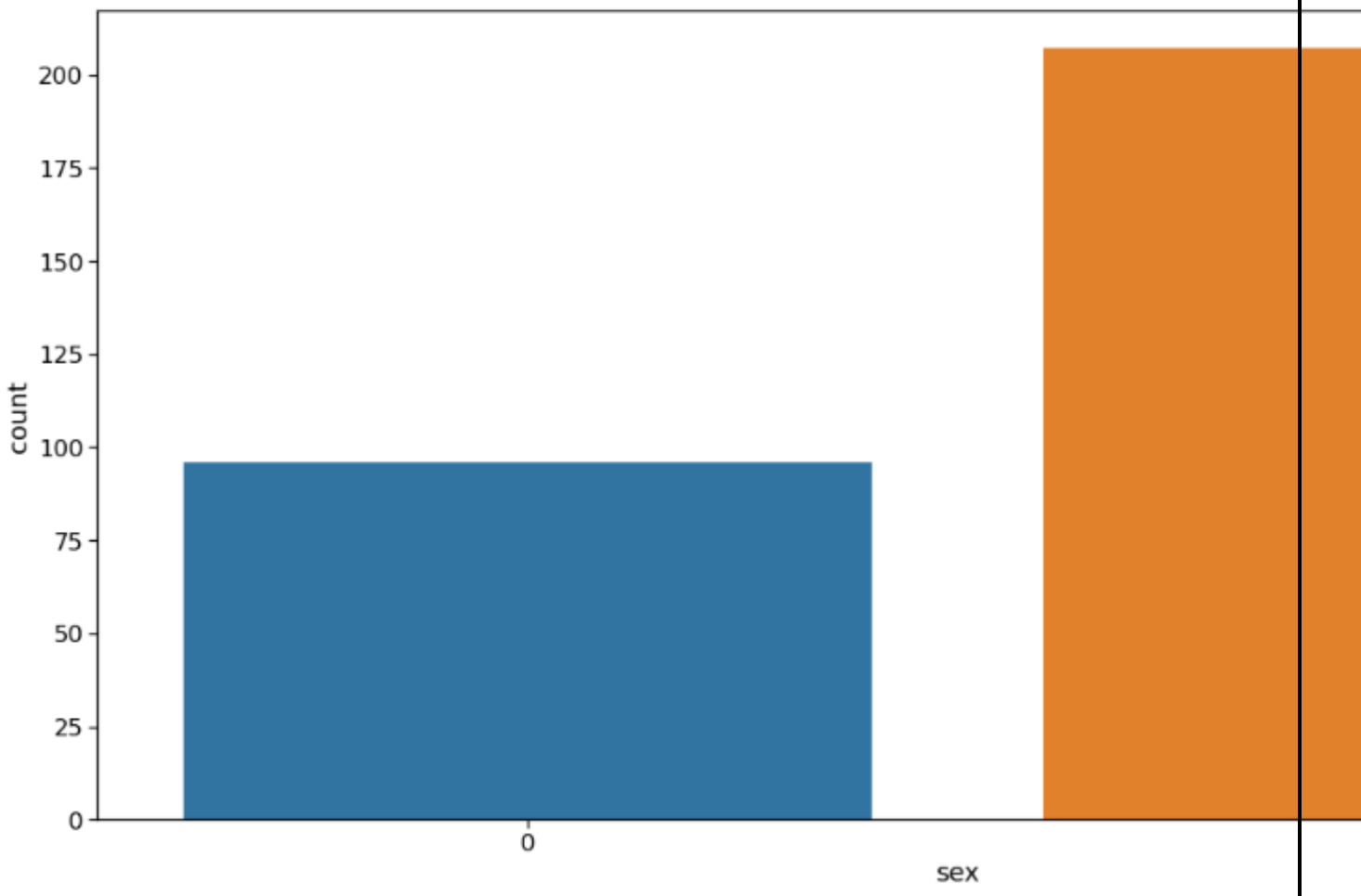


```
colors = ['red','green','yellow']
explode = [0,0,0.1]
plt.figure(figsize=(10,10))
sns.set_context('notebook',font_scale = 1.2)
plt.pie([len(Young),len(Middle),len(Elder)],labels=['young ages','middle ages','elderly ages'],explode=explo
plt.tight_layout()
```



```
plt.figure(figsize=(18,9))
sns.set_context('notebook',font_scale = 1.5)
sns.countplot(df['sex'])
plt.tight_layout()
```

```
C:\Users\rajes\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the fo
rg: x. From version 0.12, the only valid positional argument will be `data`, and passing other a
yword will result in an error or misinterpretation.
  warnings.warn(
```



Acuuracy Table:

Algorithm	Accuracy
Logistic Regression	89.00
SVM	77
Naïve Bayes	54.3636
Random Forest	85

Table 1: Accuracy metrics.

9. CONCLUSION AND FUTURE SCOPE

The study of Artificial Neural Network helped to detect the Sentimental analysis in the IMDB Movie reviews. For Sentimental Analysis detection, our system attained 89% accuracy for Multi-Layer Perceptron. There are many ways for future enhancements, but here we mention particularly two assuring ones. The first method we used is Artificial Neural Network to identify the Sentimental Analysis and Batch processing to speed up the training process. The second method is automatically styling the multiple species thereby, stepping up the accuracy.

10. ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my Guide Lakshmi Lalitha Vuyyuru as well as our HOD Hari Kiran Vege who gave me the golden opportunity to do this wonderful project.

11. REFERENCES

[1] B. Pang, L. Lee, and S. Vaithyanathan. “Thumbs up?: sentiment classification using machine learning techniques.” In Proceedings of the ACL-02 conference on Empirical methods in natural language processingVolume 10, Association for Computational Linguistics, pp. 79-86. 2002.

[2] A. Tripathy, A. Agrawal, and S.K. Rath. “Classification of sentiment reviews using n-gram machine learning approach.” Expert Systems with Applications, Vol. 57, pp. 117-126. 2016.

[3] A. G. Prasad, S. Sanjana, S. M. Bhat, and B. S. Harish. “Sentiment analysis for sarcasm detection on streaming short text data.” In Knowledge Engineering and Applications (ICKEA), 2017, 2nd International Conference on, pp. 1-5. IEEE, 2017.

[4] P. Melville, W. Gryc, and R. D. Lawrence. “Sentiment analysis of blogs by combining lexical knowledge with text classification.” In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp. 1275-1284. 2009.

[5] Jianqiang Z, Xiaolin G, Xuejun Z (2018) Deep convolution neural networks for twitter sentiment analysis. IEEE Access 6:23253–23260.

[6] Iqbal F et al (2019) A hybrid framework for sentiment analysis using genetic algorithm based feature reduction. IEEE Access 7:14637–14652.

[7] C. D. Santos and M. Gatti. “Deep convolutional neural networks for sentiment analysis of short texts.” In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, 2014, pp. 69-78.

[8] A. Ortigosa, J. M. Martín, and R. M. Carro. “Sentiment analysis in Facebook and its application to e-learning.” Computers in Human Behavior Vol. 31, pp.527-541. 2014.

[9] R. Ahmad, A. Pervaiz, P. Mannan, and F. Zaffar. “Aspect Based Sentiment Analysis for Large Documents with Applications to US Presidential Elections 2016.” Social Technical and Social Inclusion Issues (SIGSI), 2017, pp. 13.

[10] H. M. Kumar, B. S. Harish, S. V. Kumar, and V. N. Aradhya. “Classification of sentiments in short-text: an approach using mSMTP measure”. In Proceedings of the 2nd International Conference on Machine Learning and Soft Computing. pp. 145-150. ACM. 2018.

[11] E. Cambria, D. Olsher, and D. Rajagopal, “SenticNet 3: a common and common-sense knowledge base for cognition-driven sentiment analysis,” Twenty-eighth AAAI conference on . . . , pp. 1515–1521, 2014.

[12] M. E. Moussa, E. H. Mohamed, and M. H. Haggag. “A survey on Opinion Summarization Techniques for Social Media.” Future Computing and Informatics Journal (2018). In press.

[13] I. Hemalatha, G. P. S. Varma, and A. Govardhan. “Preprocessing the informal text for efficient sentiment analysis.” International Journal of Emerging Trends & Technology in Computer Science (IJETTCS) 1, no. 2: pp.58-61. 2012.

[14] A. S. Manek, P. D. Shenoy, M. C. Mohan, and K. R. Venugopal. “Aspect term extraction for sentiment analysis in large movie reviews using Gini Index feature selection method and SVM classifier.” World Wide Web Vol. 20, no. 2, pp.135-154. 2017.

[15] A. Kennedy and D. Inkpen. “Sentiment classification of movie reviews using contextual valence shifters.” Computational intelligence, Vol. 22, no. 2, .pp.110-125. 2006.