PROJECT BASED REPORT

By

Sentiment Analysis Using Deep Learning

Submitted in partial fulfilment of the

Requirements for the award of the Degree of

Bachelor of Technology

In

Computer Science & Engineering

By

STUDENT ID	NAME
170030974	P. MANI PRASANTH
170030975	P. SRI VINAY TEJA
170031093	M. RAJ KUMAR
170031099	R. DINESH

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(DST-FIST Sponsored Department)

K L University

Green Fields, Vaddeswaram, Guntur District-522 502

2019-2020

K L University

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(DST-FIST Sponsored Department)

CERTIFICATE

This is to certify that this project based report entitled **“Sentiment Analysis** Using Deep Learning**”** is a bonafide work done by 170030974 P. MANI PRASANTH, 170030975 P. SRI VINAY TEJA, 170031093 M. RAJ KUMAR, 170031099 R. DINESH in partial fulfillment of the requirement for the award of degree in **BACHELOR OF TECHNOLOGY** in **Computer Science and Engineering** during the academic year 2019-2020.

Faculty In Charge	**Head of the Department**
Dr.DEBRUP BANERJEE	**Mr.V.Hari Kiran**

K L University

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(DST-FIST Sponsored Department)

DECLARATION

We hereby declare that this project based report entitled **“Sentiment Analysis** Using Deep Learning**”** has been prepared by us in partial fulfillment of the requirement for the award of degree “**BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING**” during the academic year 2019-2020.

I also declare that this project skilling report is of our own effort and it has not been submitted to any other university for the award of any degree.

Date: 23-11-2019

Place: Vaddeswaram

ACKNOWLEDGEMENT

My sincere thanks to **Dr.DEBRUP BANERJEE** for rendering outstanding support throughout the project for the successful completion of the work.

We express our gratitude to **Mr.V.Harikiran,** 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 projec

t work.

We would like to place on record the deep sense of gratitude to the honourable Vice Chancellor, K L University for providing the necessary facilities to carry the concluded project work.

Last but not the least, we thank all Teaching and Non-Teaching Staff of our department and especially my classmates and my friends for their support in the completion of our project work.

P. MANI PRASANTH	170030974
P. SRI VINAY TEJA	170030975
M. RAJ KUMAR	170031093
R. DINESH	170031099

TABLE OF CONTENTS

ABSTRACT

In recent years the deep learning techniques used in Sentiment Analysis. Sentiment analysis is one of the most researched areas in natural language processing. Natural language processing has a wide range of applications like voice recognition, machine translation, movie review, aspect oriented product analysis, sentiment analysis and text classification like email categorization and spam filtering. The conventional methods used for sentiment analysis is lexicon based processing. However, with the advancements in the field of artificial intelligence, the machine learning algorithms started to play a major role in sentiment analysis applications. Currently deep learning technique is the latest hotspot being used for predicting the sentiments. Several research works have been carried out in the Natural Language Processing (NLP) using the deep learning methods. The most popular deep learning methods employed includes Convolution Neural Network (CNN) and Recurrent Neural Network (RNN) particularly the Long Short Term Memory (LSTM). These techniques are used in combination or as stand-alone based on the domain area of application. The focus of this survey is on the various flavors of the deep learning methods used in different applications of sentiment analysis at sentence level and aspect/target level.

1. INTRODUCTION

Sentiment analysis means the process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether the writer's attitude towards a particular topic, product, etc. is positive, negative, or neutral.

Deep Learning means a class of machine learning algorithms that uses multiple layers to progressively extract higher level features from the raw input. For example, in image processing, lower layers may identify edges, while higher layers may identify the concepts relevant to a human such as digits or letters or faces.

The best businesses understand the sentiment of their customers what people are saying, how they're saying it, and what they mean. Customer sentiment can be found in tweets, comments, reviews, or other places where people mention your brand. Sentiment Analysis is the domain of understanding these emotions with software, and it's a must-understand for developers and business leaders in a modern workplace.

As with many other fields, advances in deep learning have brought sentiment analysis into the foreground of cutting-edge algorithms. Today we use natural language processing, statistics, and text analysis to extract, and identify the sentiment of text into positive, negative, or neutral categories.

2. Literature Survey

2.1. About Naive Bayes Classifier:

In machine learning, naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong independence assumptions between the features. They are among the simplest Bayesian network models.

Naïve Bayes has been studied extensively since the 1960s. It was introduced into the text retrieval community in the early 1960s, and remains a popular (baseline) method for text categorization, the problem of judging documents as belonging to one category or the other with word frequencies as the features. With appropriate pre-processing, it is competitive in this domain with more advanced methods including support vector machines. It also finds application in automatic medical diagnosis.

Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression, which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers.

3. METHODOLOGY

Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

Naïve Bayes Classifier Formula:

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

Fig.3.1

Naive Bayes Classifier Formula

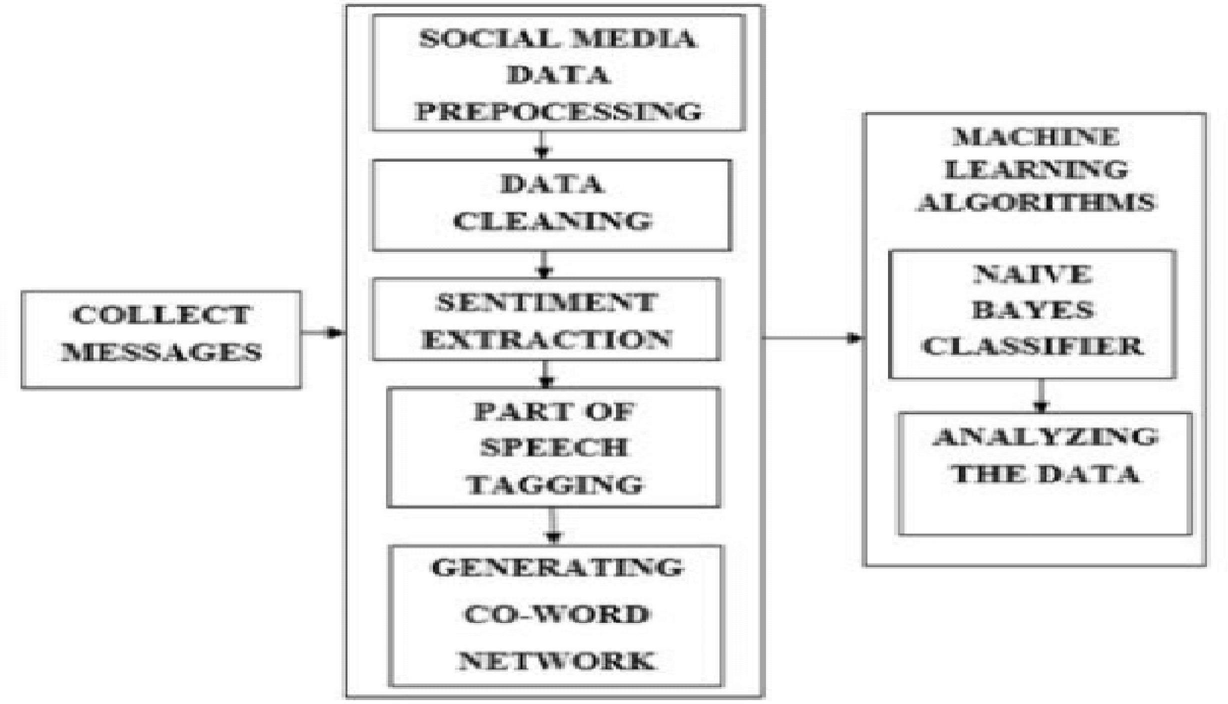


Fig.3.1

Block Diagram Naive Bayes Classifier

3.2.Gaussian Naive Bayes:

When the predictors take up a continuous value and are not discrete, we assume that these values are sampled from a gaussian distribution.

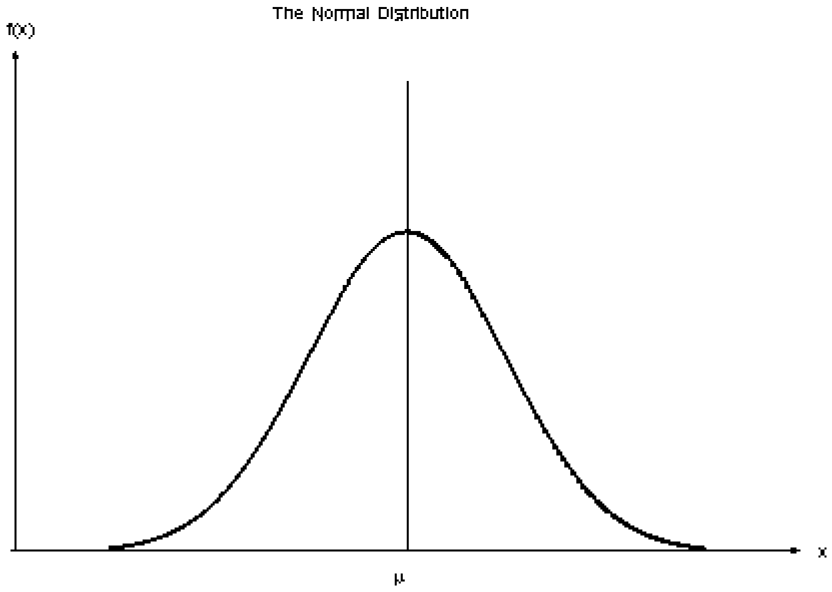


Fig.3.2

Gaussian Distribution(Normal Distribution)

Since the way the values are present in the dataset changes, the formula for conditional probability changes to,

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

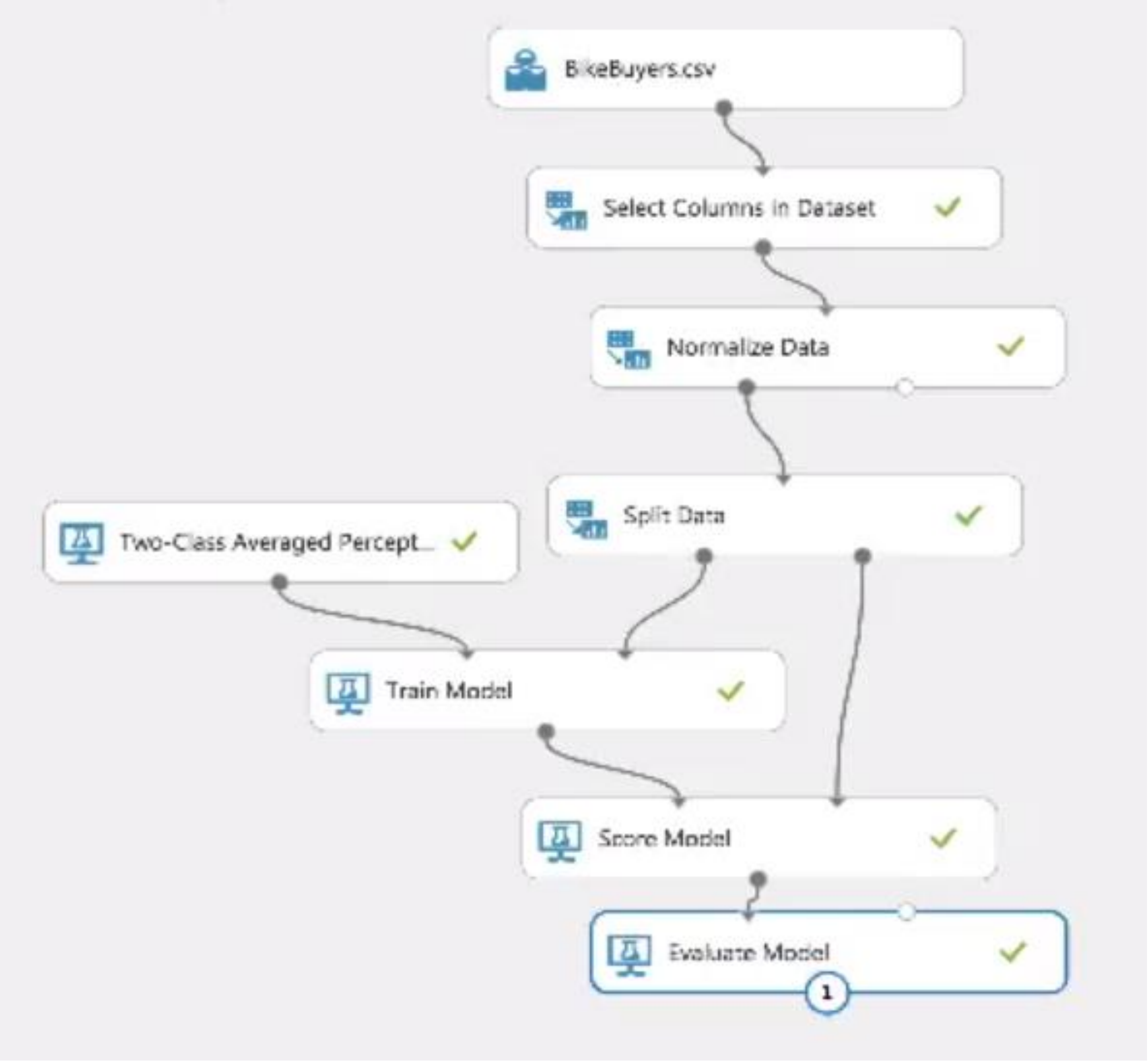


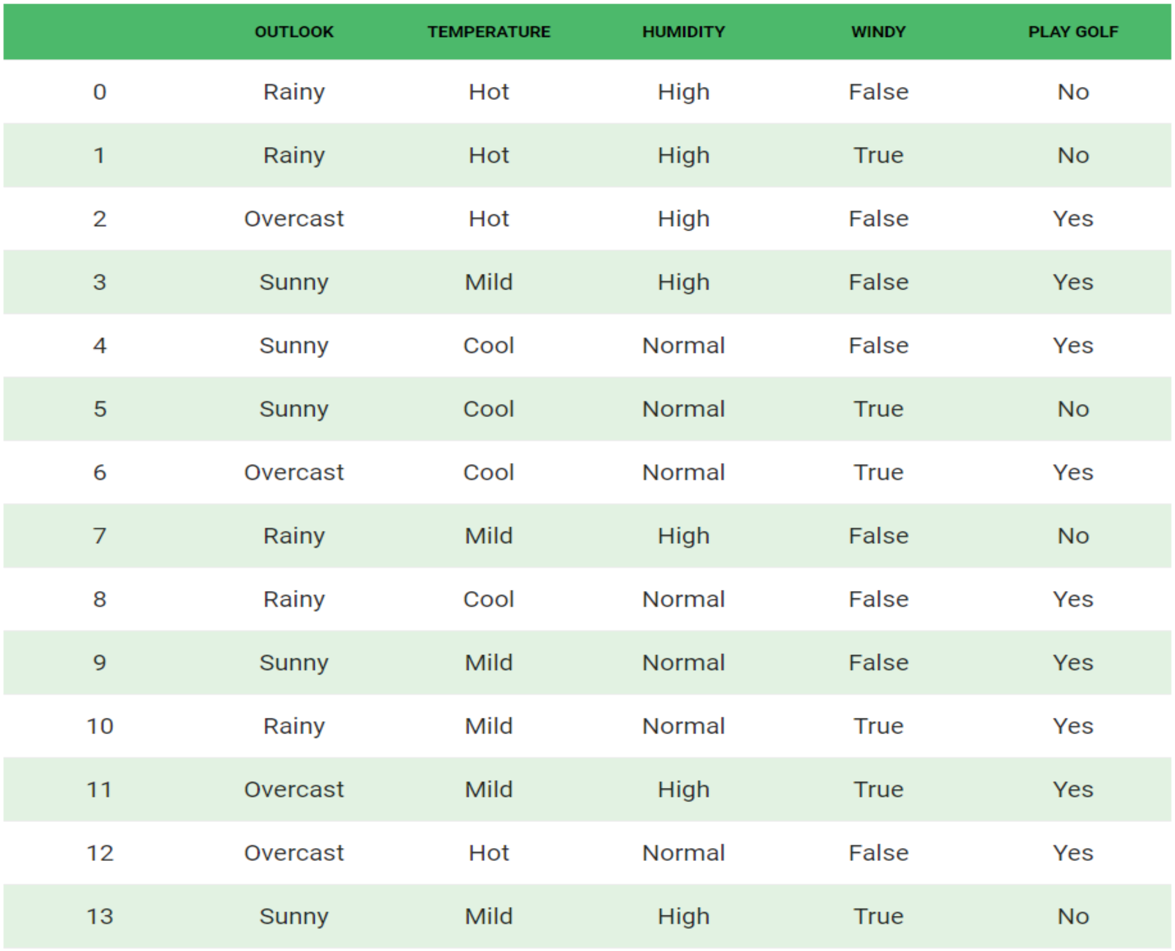
Fig.3.3

Naïve Bayes Classification flowchart in Azure

To start with, let us consider a dataset.

Consider a fictional dataset that describes the weather conditions for playing a game of golf. Given the weather conditions, each tuple classifies the conditions as fit(“Yes”) or unfit(“No”) for plaing golf.

Here is a tabular representation of our dataset.



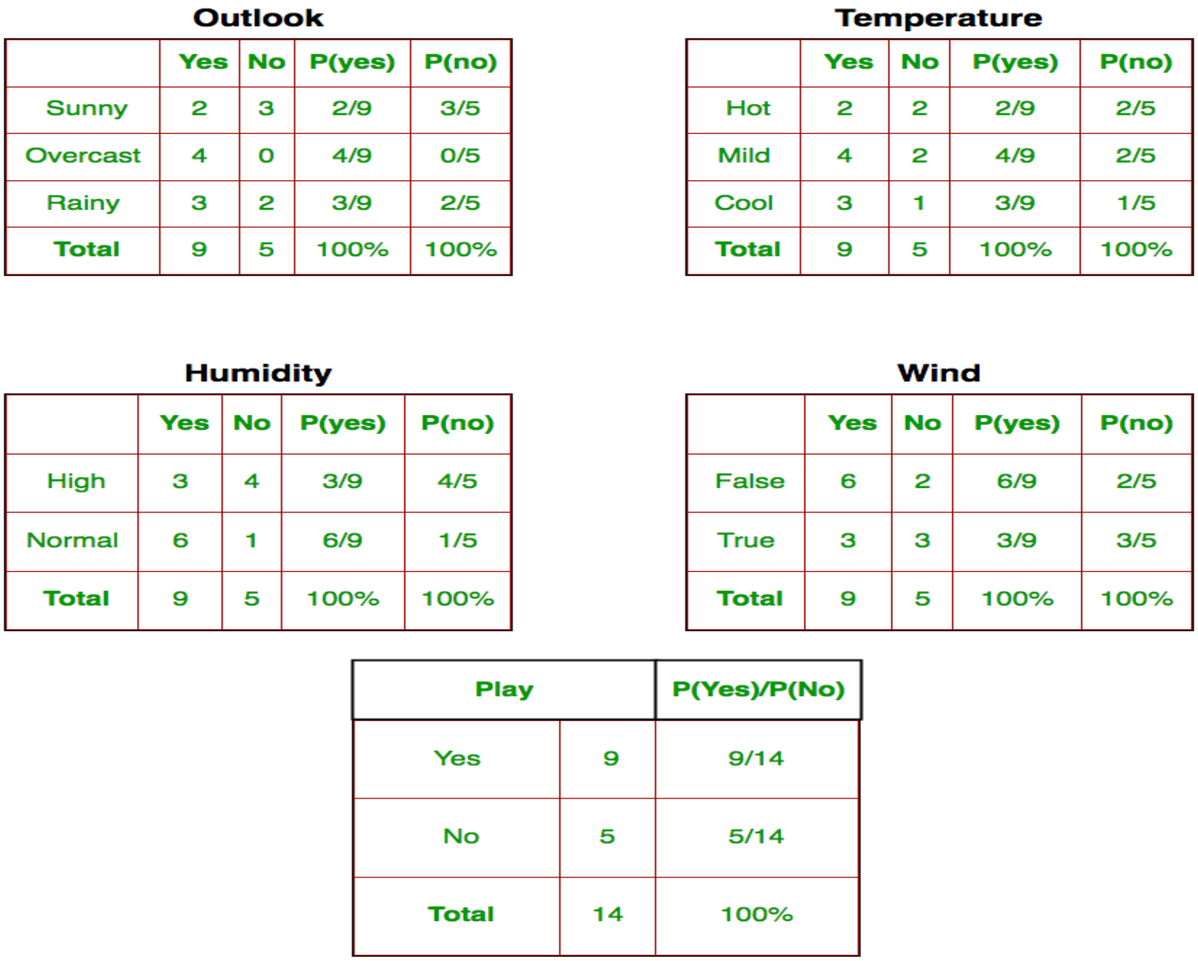
	OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY GOLF
0	Rainy	Hot	High	False	No
1	Rainy	Hot	High	True	No
2	Overcast	Hot	High	False	Yes
3	Sunny	Mild	High	False	Yes
4	Sunny	Cool	Normal	False	Yes
5	Sunny	Cool	Normal	True	No
6	Overcast	Cool	Normal	True	Yes
7	Rainy	Mild	High	False	No
8	Rainy	Cool	Normal	False	Yes
9	Sunny	Mild	Normal	False	Yes
10	Rainy	Mild	Normal	True	Yes
11	Overcast	Mild	High	True	Yes
12	Overcast	Hot	Normal	False	Yes
13	Sunny	Mild	High	True	No

Fig. 3.4

Data Set to play golf or not

The dataset is divided into two parts, namely, feature matrix and the response vector.

1. Feature matrix contains all the vectors(rows) of dataset in which each vector consists of the value of dependent features. In above dataset, features are ‘Outlook’, ‘Temperature’, ‘Humidity’ and ‘Windy’.
2. Response vector contains the value of class variable(prediction or output) for each row of feature matrix. In above dataset, the class variable name is ‘Play golf’.



Outlook

	Yes	No	P(yes)	P(no)
Sunny	2	3	2/9	3/5
Overcast	4	0	4/9	0/5
Rainy	3	2	3/9	2/5
Total	9	5	100%	100%

Temperature

	Yes	No	P(yes)	P(no)
Hot	2	2	2/9	2/5
Mild	4	2	4/9	2/5
Cool	3	1	3/9	1/5
Total	9	5	100%	100%

Humidity

	Yes	No	P(yes)	P(no)
High	3	4	3/9	4/5
Normal	6	1	6/9	1/5
Total	9	5	100%	100%

Wind

	Yes	No	P(yes)	P(no)
False	6	2	6/9	2/5
True	3	3	3/9	3/5
Total	9	5	100%	100%

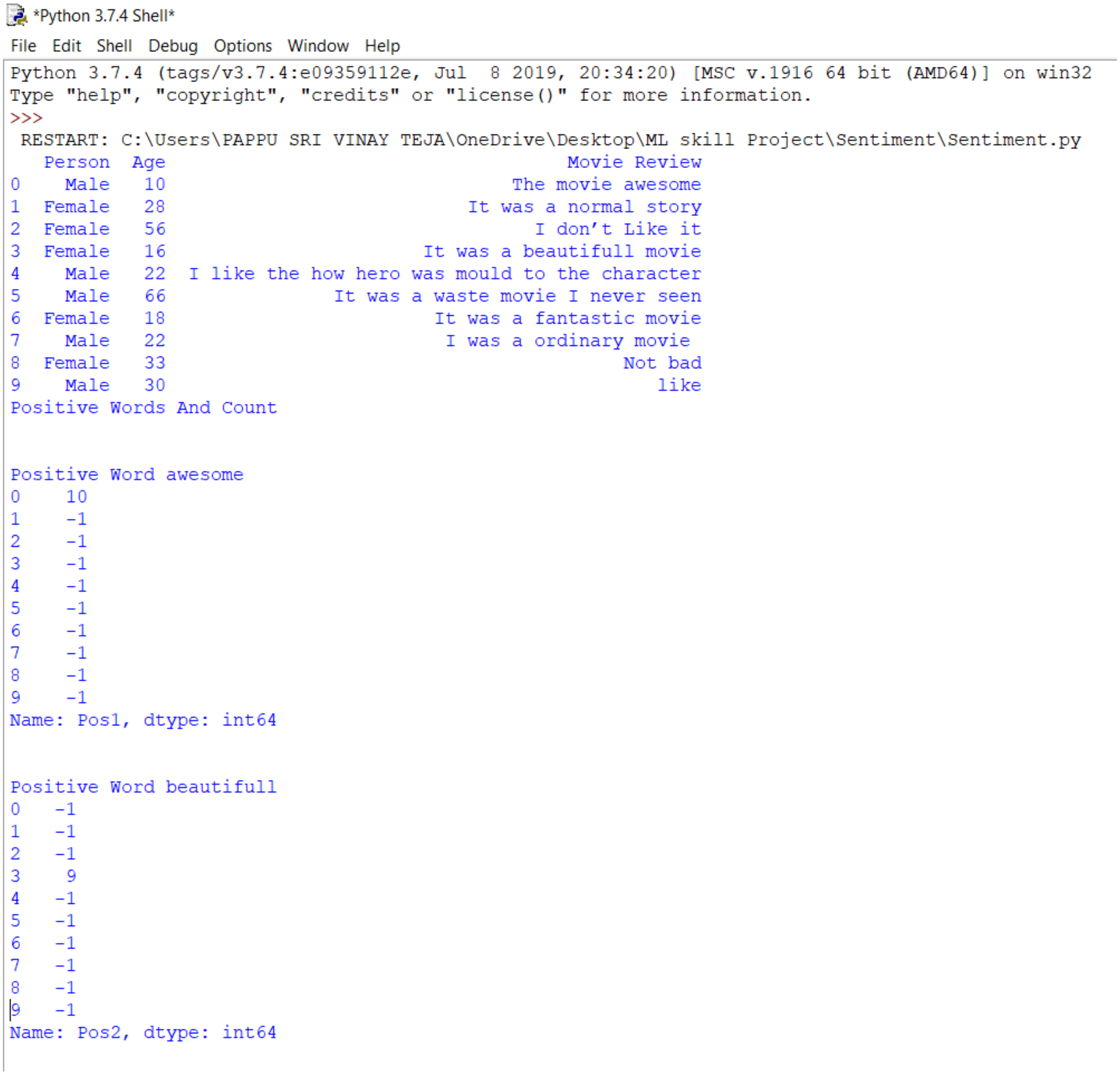
Play		P(Yes)/P(No)
Yes	9	9/14
No	5	5/14
Total	14	100%

Fig.3.5

Separated data set into two parts, namely, feature matrix and the response vector.

4. RESULT AND DISCUSSION

4.1.1 : Module 1 output:

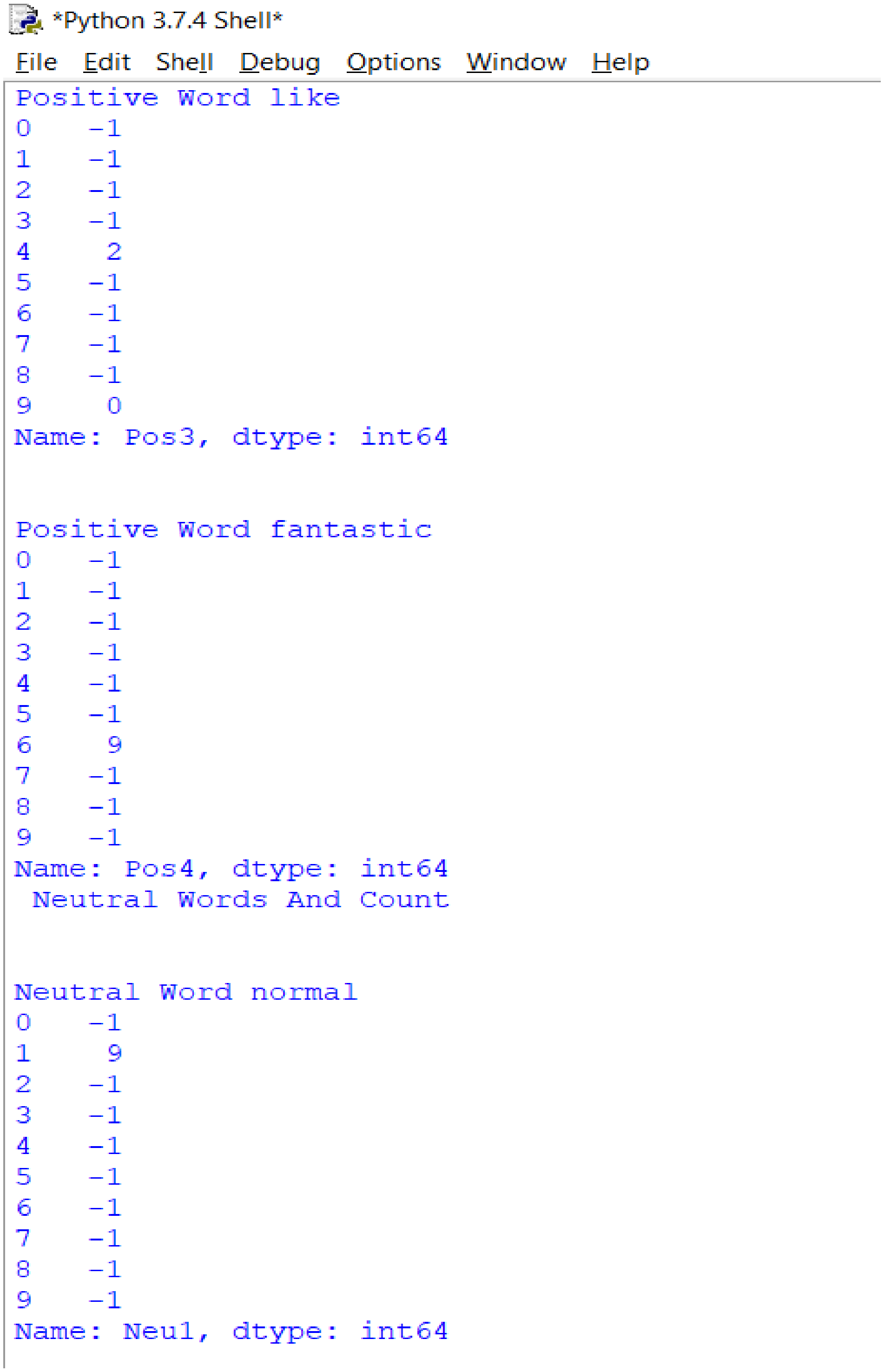


```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
 RESTART: C:\Users\PAPPU SRI VINAY TEJA\OneDrive\Desktop\ML skill Project\Sentiment\Sentiment.py
   Person  Age                                      Movie Review
0    Male   10                                 The movie awesome
1  Female   28                             It was a normal story
2  Female   56                                   I don’t Like it
3  Female   16                          It was a beautifull movie
4    Male   22  I like the how hero was mould to the character
5    Male   66                    It was a waste movie I never seen
6  Female   18                          It was a fantastic movie
7    Male   22                           I was a ordinary movie 
8  Female   33                                           Not bad
9    Male   30                                              like
Positive Words And Count


Positive Word awesome
0    10
1    -1
2    -1
3    -1
4    -1
5    -1
6    -1
7    -1
8    -1
9    -1
Name: Pos1, dtype: int64


Positive Word beautifull
0   -1
1   -1
2   -1
3    9
4   -1
5   -1
6   -1
7   -1
8   -1
9   -1
Name: Pos2, dtype: int64
```

Fig.4.1

This figure show the data set and the number of positive word from the movie reviews

4.1.2 : Module 1,2 output:

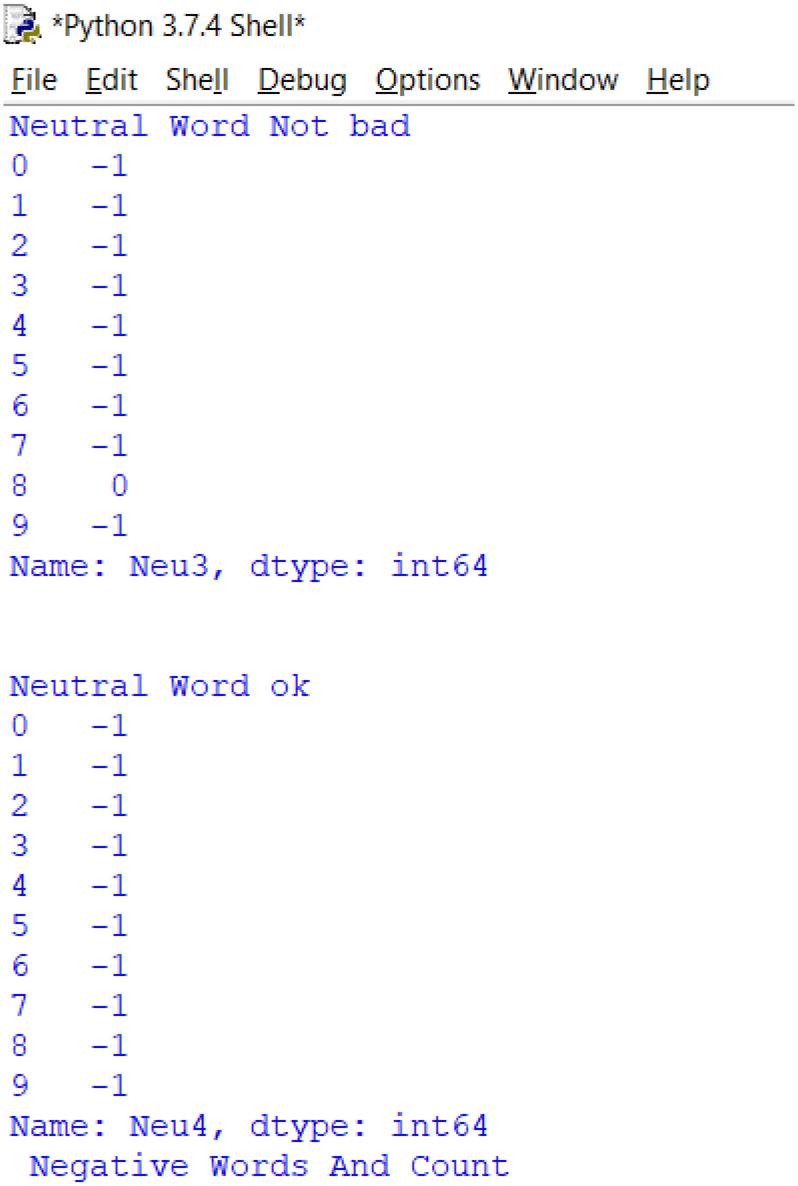


```
Positive Word like
0    -1
1    -1
2    -1
3    -1
4     2
5    -1
6    -1
7    -1
8    -1
9     0
Name: Pos3, dtype: int64


Positive Word fantastic
0    -1
1    -1
2    -1
3    -1
4    -1
5    -1
6     9
7    -1
8    -1
9    -1
Name: Pos4, dtype: int64
 Neutral Words And Count


Neutral Word normal
0    -1
1     9
2    -1
3    -1
4    -1
5    -1
6    -1
7    -1
8    -1
9    -1
Name: Neu1, dtype: int64
```

Fig.4.1.2

This figure show the number of positive, neutral word from the movie reviews

4.1.3 : Module 2 output:

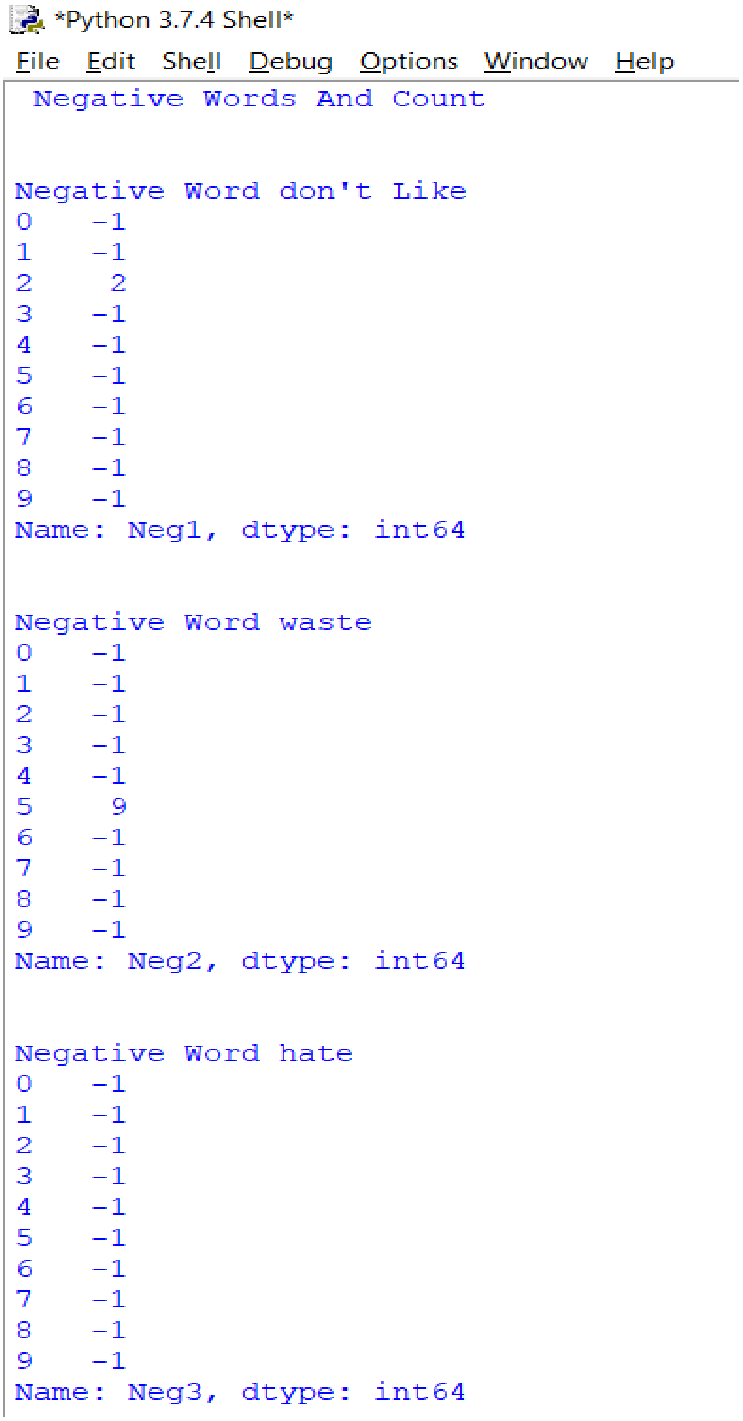


```
Neutral Word Not bad
0   -1
1   -1
2   -1
3   -1
4   -1
5   -1
6   -1
7   -1
8    0
9   -1
Name: Neu3, dtype: int64


Neutral Word ok
0   -1
1   -1
2   -1
3   -1
4   -1
5   -1
6   -1
7   -1
8   -1
9   -1
Name: Neu4, dtype: int64
 Negative Words And Count
```

Fig.4.1.3

This figure show the number of neutral word from the movie reviews

4.1.4 : Module 3 output:

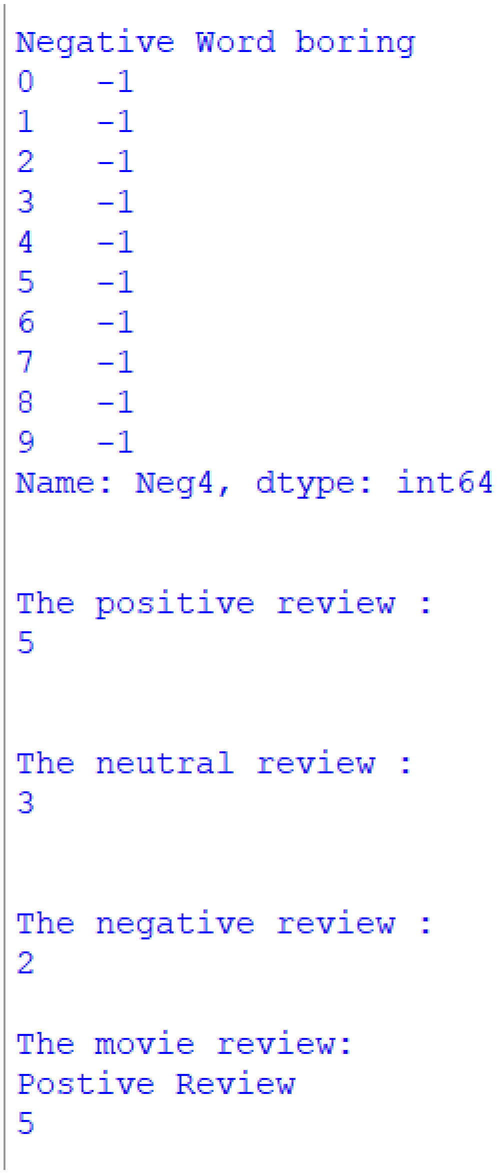


```
 Negative Words And Count


Negative Word don't Like
0    -1
1    -1
2     2
3    -1
4    -1
5    -1
6    -1
7    -1
8    -1
9    -1
Name: Neg1, dtype: int64


Negative Word waste
0    -1
1    -1
2    -1
3    -1
4    -1
5     9
6    -1
7    -1
8    -1
9    -1
Name: Neg2, dtype: int64


Negative Word hate
0    -1
1    -1
2    -1
3    -1
4    -1
5    -1
6    -1
7    -1
8    -1
9    -1
Name: Neg3, dtype: int64
```

Fig.4.1.4

This figure show the number of negtive word from the movie reviews

4.1.5 : Module 3 output:



```
Negative Word boring
0    -1
1    -1
2    -1
3    -1
4    -1
5    -1
6    -1
7    -1
8    -1
9    -1
Name: Neg4, dtype: int64


The positive review :
5


The neutral review :
3


The negative review :
2

The movie review:
Postive Review
5
```

Fig.4.1.5

This figure show the number of negtive word and final result of movie

<u>Fig.4.1.6 : Module 4 output:</u>

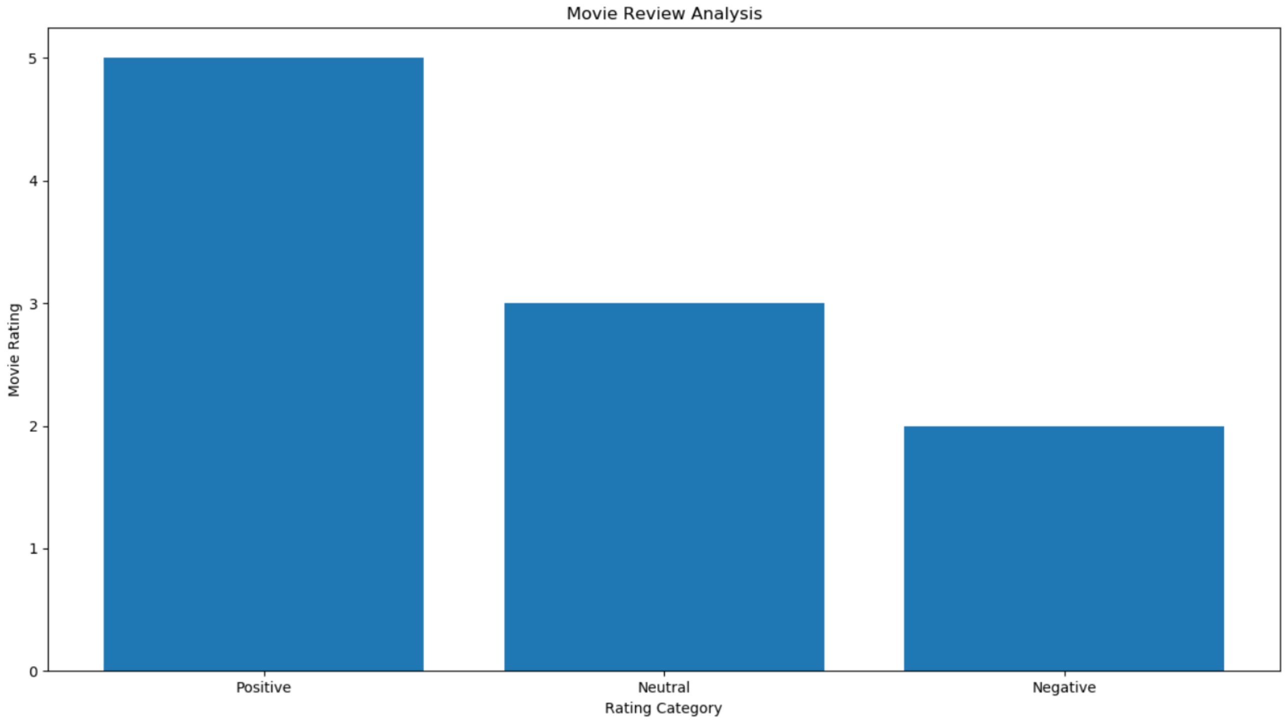


Fig.4.1.6

The movie review rating bar graph. The x axis shows the Rating category and y axis shoe the Rating of the movie give by people

Fig.4.1.7 : Module 4 output:

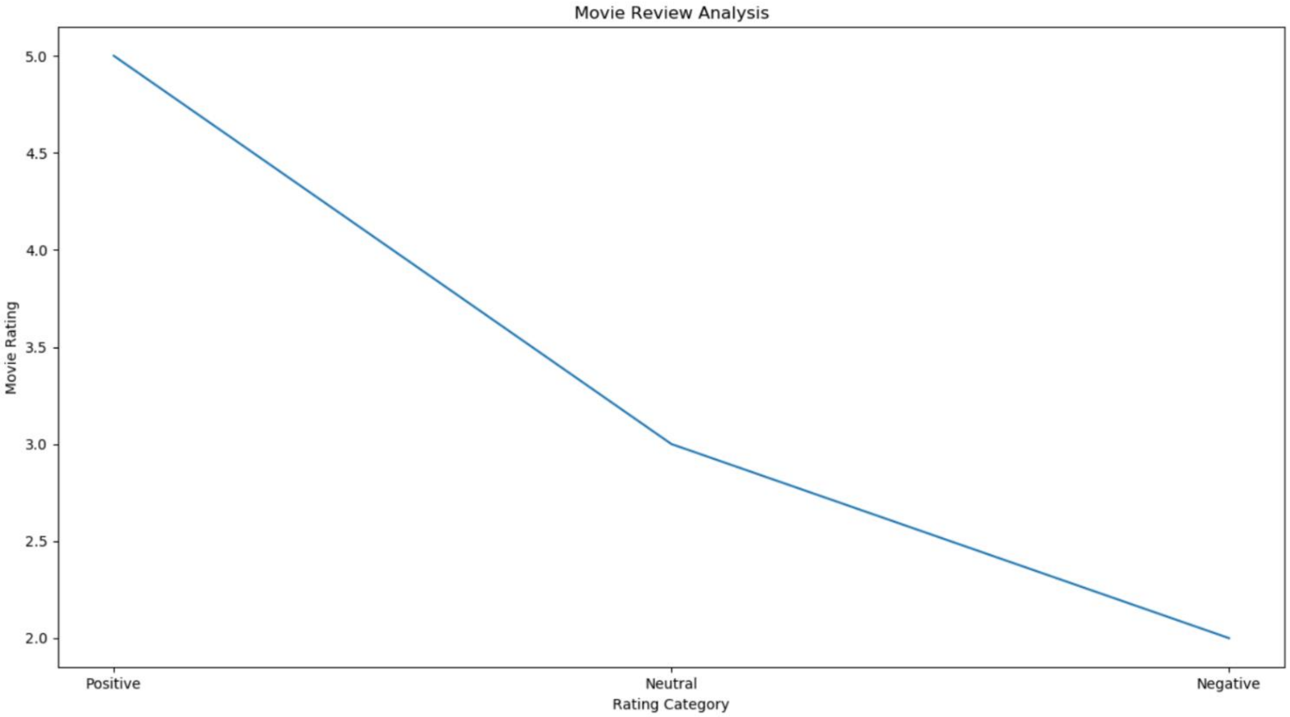
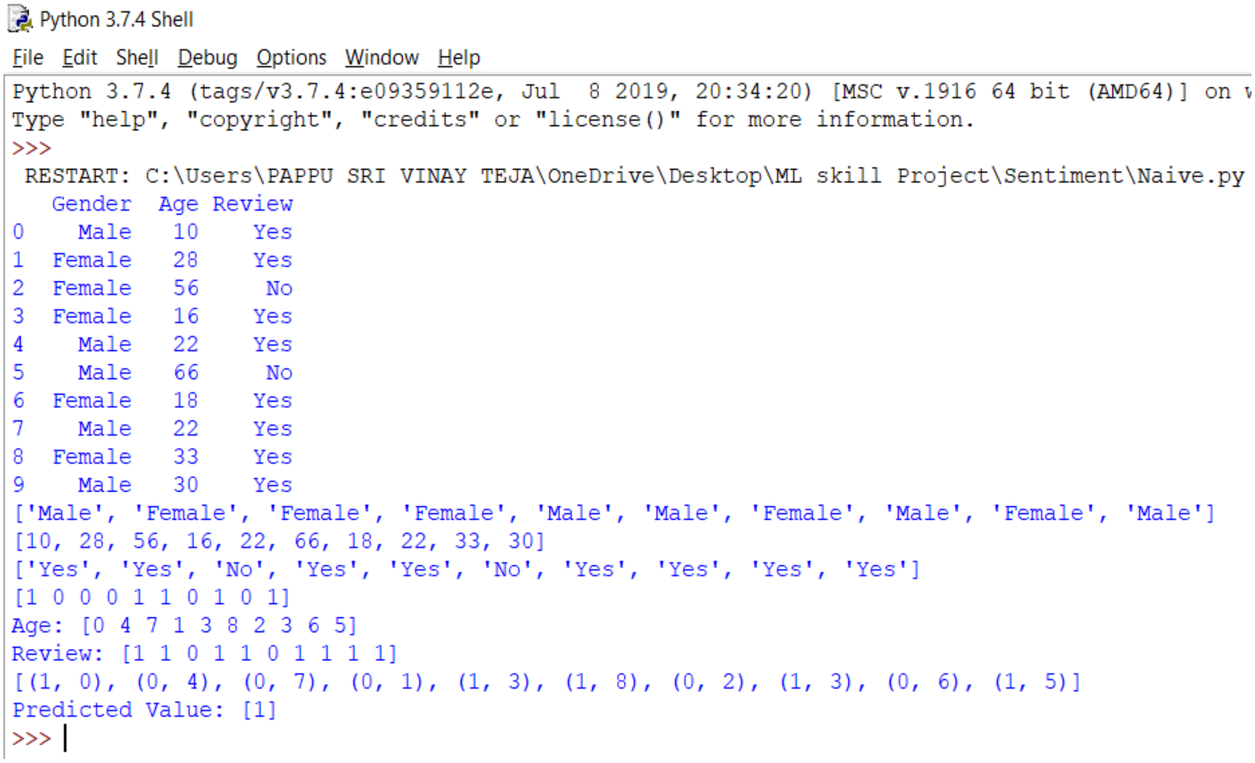


Fig.4.1.7

The movie review rating bar graph. The x axis shows the Rating category and y axis shoe the Rating of the movie give by people

Fig.4.2.1 : Module 5 output:



```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on w
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
 RESTART: C:\Users\PAPPU SRI VINAY TEJA\OneDrive\Desktop\ML skill Project\Sentiment\Naive.py
   Gender  Age Review
0    Male   10    Yes
1  Female   28    Yes
2  Female   56     No
3  Female   16    Yes
4    Male   22    Yes
5    Male   66     No
6  Female   18    Yes
7    Male   22    Yes
8  Female   33    Yes
9    Male   30    Yes
['Male', 'Female', 'Female', 'Female', 'Male', 'Male', 'Female', 'Male', 'Female', 'Male']
[10, 28, 56, 16, 22, 66, 18, 22, 33, 30]
['Yes', 'Yes', 'No', 'Yes', 'Yes', 'No', 'Yes', 'Yes', 'Yes', 'Yes']
[1 0 0 0 1 1 0 1 0 1]
Age: [0 4 7 1 3 8 2 3 6 5]
Review: [1 1 0 1 1 0 1 1 1 1]
[(1, 0), (0, 4), (0, 7), (0, 1), (1, 3), (1, 8), (0, 2), (1, 3), (0, 6), (1, 5)]
Predicted Value: [1]
>>> |
```

Fig.4.2.1

Naïve Bayes Classifier calculates and predicted result of the required attributes

5.CONCLUSION AND FUTURE WORK

Naive Bayes algorithms are mostly used in sentiment analysis, spam filtering, recommendation systems etc. They are fast and easy to implement but their biggest disadvantage is that the requirement of predictors to be independent. In most of the real life cases, the predictors are dependent, this hinders the performance of the classifier.

5.1:CONCLUSION BASED ON THE RESULT:

In this system the we find the feeling on the movie based on the review it sentiment analysis. From the data set we find the specific from the review and compare the word with lexicon data words we separate the data.

The divided data is give to Naïve Bayes Classifier and train, test and predict the review For this calculation we divided the data set into required subset.

Finally, we find the probability to the every sub set and predict the result.

6. REFERENCES:

1. https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c#:~:targetText=Conclusion%3A,of%20predictors%20to%20be%20independent.
2. https://www.geeksforgeeks.org/naive-bayes-classifiers/#:~:targetText=Naive%20Bayes%20classifiers%20are%20a,is%20independent%20of%20each%20other.
3. https://algorithmia.com/blog/introduction-sentiment-analysis#:~:targetText=As%20with%20many%20other%20fields,%2C%20negative%2C%20or%20neutral%20categories.
4. https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.1253
5. https://www.datacamp.com/community/tutorials/naive-bayes-scikit-learn
6. https://www.youtube.com/watch?v=CPqOCI0ahss

7. APPENDIX

Code:

a) Code to find the number of positive, negative, neutral words are in the given dataset.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
data=pd.read_csv('C:\\Users\\PAPPU SRI VINAY
TEJA\\OneDrive\\Desktop\\ML skill
Project\\Sentiment\\Dataframe1.csv')
print(data)
#positive words and count
sub1 ='awesome'
sub2='beautifull'
sub3='like'
sub4='fantastic'
data["Pos1"]= data["Movie Review"].str.find(sub1)
print("Positive Words And Count")
print("\n")
print("Positive Word awesome")
print(data['Pos1'])
data["Pos2"]= data["Movie Review"].str.find(sub2)
print("\n")
print("Positive Word beautifull")
print(data['Pos2'])
data["Pos3"]= data["Movie Review"].str.find(sub3)
print("\n")
print("Positive Word like")
print(data['Pos3'])
data["Pos4"]= data["Movie Review"].str.find(sub4)
print("\n")
print("Positive Word fantastic")
print(data['Pos4'])
```

```
Pos1=list(data['Pos1'])
Pos2=list(data['Pos2'])
Pos3=list(data['Pos3'])
Pos4=list(data['Pos4'])
pos_count= 0
for num in Pos1:
    if num >= 0:
        pos_count += 1
for num in Pos2:
    if num >= 0:
        pos_count += 1
for num in Pos3:
    if num >= 0:
        pos_count += 1
for num in Pos4:
    if num >= 0:
        pos_count += 1

#neutral words and count
su1 ='normal'
su2='ordinary'
su3='Not bad'
su4='ok'
data["Neu1"]= data["Movie Review"].str.find(su1)
print(" Neutral Words And Count")
print("\n")
print("Neutral Word normal")
print(data['Neu1'])
data["Neu2"]= data["Movie Review"].str.find(su2)
print("\n")
print("Netural Word ordinary")
print(data['Neu2'])
data["Neu3"]= data["Movie Review"].str.find(su3)
print("\n")
print("Neutral Word Not bad")
print(data['Neu3'])
data["Neu4"]= data["Movie Review"].str.find(su4)
print("\n")
print("Neutral Word ok")
print(data['Neu4'])
Neu1=list(data['Neu1'])
Neu2=list(data['Neu2'])
Neu3=list(data['Neu3'])
Neu4=list(data['Neu4'])
neu_count= 0
for num in Neu1:
    if num >= 0:
        neu_count += 1
for num in Neu2:
    if num >= 0:
        neu_count += 1
for num in Neu3:
    if num >= 0:
        neu_count += 1
for num in Neu4:
    if num >= 0:
        neu_count += 1


#negative words and count
s1 ='don’t Like'
s2='waste'
s3='hate'
s4='boring'
data["Neg1"]= data["Movie Review"].str.find(s1)
print(" Negative Words And Count")
print("\n")
print("Negative Word don't Like")
print(data['Neg1'])
data["Neg2"]= data["Movie Review"].str.find(s2)
```

```
print("\n")
print("Negative Word waste")
print(data['Neg2'])
data["Neg3"]= data["Movie Review"].str.find(s3)
print("\n")
print("Negative Word hate")
print(data['Neg3'])
data["Neg4"]= data["Movie Review"].str.find(s4)
print("\n")
print("Negative Word boring")
print(data['Neg4'])
Neg1=list(data['Neg1'])
Neg2=list(data['Neg2'])
Neg3=list(data['Neg3'])
Neg4=list(data['Neg4'])
neg_count= 0
for num in Neg1:
    if num >= 0:
        neg_count += 1
for num in Neg2:
    if num >= 0:
        neg_count += 1
for num in Neg3:
    if num >= 0:
        neg_count += 1
for num in Neg4:
    if num >= 0:
        neg_count += 1
print("\n")
print("The positive review :")
print(pos_count)
print("\n")
print("The neutral review :")
print(neu_count)
print("\n")
print("The negative review :")
print(neg_count)
x=['Positive','Neutral','Negative']
y=list=[pos_count,neu_count,neg_count]
print("\nThe movie review:")
if(max(y)==pos_count):
    print("Postive Review")
    print(pos_count)
elif(max(y)==neu_count):
    print("Neutral Review")
    print(neu_count)
else:
    print("Negative Review")
    print(neg_count)
plt.bar(x,y)
plt.title('Movie Review Analysis')
plt.xlabel('Rating Category')
plt.ylabel('Movie Rating')
plt.show()
plt.plot(x,y)
plt.title('Movie Review Analysis')
plt.xlabel('Rating Category')
plt.ylabel('Movie Rating')
plt.show()
```

b) Code to do Naïve Bayes classification.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import preprocessing
data=pd.read_csv('C:\\Users\\PAPPU SRI VINAY
TEJA\\OneDrive\\Desktop\\ML skill
Project\\Sentiment\\Dataframe2.csv')
print(data)
```

```
Gender=list(data['Gender'])
print(Gender)
Age=list(data['Age'])
print(Age)
Review=list(data['Review'])
print(Review)
le=preprocessing.LabelEncoder()
Gender_encoder=le.fit_transform(Gender)
print(Gender_encoder)
Age_encoder=le.fit_transform(Age)
label=le.fit_transform(Review)
print("Age:",Age_encoder)
print("Review:",label)
def merge(list1, list2):
    merged_list = [(list1[i], list2[i]) for i in
range(0, len(list1))]
    return merged_list
# Driver code
list1 = Gender_encoder
list2 =  Age_encoder
Result=(merge(list1, list2))
print(Result)
from sklearn.naive_bayes import GaussianNB

#Create a Gaussian Classifier
model = GaussianNB()

# Train the model using the training sets
model.fit(Result,label)

#Predict Output
predicted= model.predict([[0,2]]) # 0:Overcast, 2:Mild
print("Predicted Value:", predicted)
```