Social media survey using decision tree and naive Bayes classification.

ROSHINI, T., SIREESHA, P.V., PARASA, D. and BANO, S.

2019

Social Media Survey using Decision Tree and Naive Bayes Classification

T. Roshini
Dept of Computer Science and Engineering
Koneru Lakshmaiah Education Foundation
Vaddeswaram, India
t.roshini608@gmail.com

P. Venkata Sireesha
Dept of Computer Science and Engineering
Koneru Lakshmaiah Education Foundation
Vaddeswaram,India
siriparimi555@gmail.com

Dhanush Parasa
Dept of Computer Science and Engineering
Koneru Lakshmaiah Education Foundation
Vaddeswaram, India
cody20734@gmail.com

Shahana Bano
Assoc Professor
Koneru Lakshmaiah Education Foundation
Vaddeswaram, India
shahanabano@icloud.com

***Abstract*—Social media is one of the most important aspects of our day to day life. For you my wonder what exactly is social media. Social media is nothing more than a website or an application that is used to create and share content among a social networking. Recent studies claim that an average person spends roughly 142 minutes per day on some form of social media. Now that may seem like a small number but considering how many people are addicted to social media might make the number far larger. Over the past few years the daily usage of social media for an average person has increased from a mere 100 minutes per day to its current 142 minutes per day. Although people around the world are spending a chunk of their day on social media platforms it is hard to identify whether such platforms are a boon or a con for mankind. Although most people argue that social media is purely a waste of time, a recent study was able to establish a conclusion that people who use social media have lower stress levels. A women who used social media several times a day scored 21% less stress levels then that of a women who had no interest of social media at all. However there are many argument out there to support that it is a bad influence among people as well. One of the most popular one being the fact that people simply are so caught up with social media that they forget the value or even how to interact with someone face to face. We weren't particularly interested in the effects of social media but we wanted to learn what types of social media platforms people preferred. Considering the fact that we live in a digital era where any data on the internet can be easily manipulated we wanted to find out how secure people felt on each social media platform. Keeping all these in mind we decided to learn more about people's approach to social media platforms.**

" How Do People React and Feel Towards Their Social Media Platforms? "

***IndexTerms*: Navie bayes classification,Decision tree**

I. INTRODUCTION

Social Media is one of the most used platforms on the internet. recent statistics have shown that there are 3.484 billion people who use social media users worldwide at the start of 2019. This was said to be a whopping 9% increase from the prior year. We can clearly see how popular and widespread social media has become.Social Media can not be linked to only one platform. In the modern day there a tons and several platforms that you can choose from. A few of the most recognized social media platforms are Facebook, Whatsapp, YouTube, Instagram, Hangouts and Gmail. These are some of the short list of the many social media platforms that are available to the world. To collect statistics on how people feel about social media we conducted a large survey that mostly consisted of students from our college and neighboring colleges as well. We created a survey that focuses on what platforms people prefer and exactly how secure they feel creating or sharing content across these platforms. Once we successfully had established a data set we took the help of **R-Programming** to easily analyse our datasets.

"R" is an programming language and an open source software for statistical computing and graphics. **"R"** programming is based on statistical programming language. It is widely used among statisticians and data miners for developing statistical software and data analysis. There are many functions that **"R"** can perform on datasets. However the most common and well used functions are Linear regression, Logistic regression, Decision tree, SVM, Naïve Bayes, KNN, K-means, and Random Forest. These may be only a few of the established functions however these are the most commonly used functions by its users. To analyze our Data we will be using two of these functions which are the ***Decision Tree*** along with ***Naïve Bayes Classification.***

II . PROCEDURE

Our study consisted of 2 major parts. In our first part we analyzed our data to identify the most common social media platform among the people. We were able to accomplish this using the help of the ***Decision Tree*** function that is inbuilt into **"R".** The second part of our project mostly focused on what social platform people found the most secure. To get accurate

results we had to use ***Naïve Bayes Classification.*** Although both of them may vary in the functionalities that they will call or possess they both will access our dataset in the same manner. Let's further go into these steps one after another.

A. Collection of Data:

In order for us to commence our study on the lifestyles of the current generation we needed data sets to work with. We needed data sets that would help us understand 2 major aspects…

i) Which social media app do they use more.

ii)And do they feel it is secure or not.

Keeping these two key aspects in mind we created an online survey using google forms for students to fill out. The survey mainly focused on a person's daily habits focusing on what types of social media apps they prefer. From these habits we ended our survey with a simple question on their views or opinions on whether which site they feel comfort and secure. Once our survey was completed we were able to output our data in the form of an Excel sheet. The sheet had all the questions placed along the x-axis while each entry was considered on the y-axis. Once we have properly saved our excel sheet on our computer we can now progress to the next stage of implementation and Analysis.

B. Importing our Meta-Data

The most important step in any process is gathering all of our pre-requirements. In this stage we must import 2 of the most important requirements into our program, the data along with the required libraries for analysis.

a) Importing our Survey Data:

We will be implementing our project with the help of ' R '. In order for us to implement all our Data-sets which are stored in a single excel sheet into our program we must undergo the following procedure. The most important part is making sure that our sheet is properly saved in the folder of our wish. From here on out we will use the setwd() inbuilt within ' R ' so that we can set the path to the folder in which our sheet is saved in. For you to get a clear and easy understanding, let's suppose that we have saved our sheet on my desktop. In such a scenario we will use the following command.

setwd("C:/Users/mylaptop/Desktop")

Once we have used the setwd() function in ' R ' the path will be properly set to the designated folder. The next step involves selecting our data sheet so that our program can further access it for future analysis. To do this we will be using the inbuilt read.csv() function. This function however will return our file to have it stored in the form of a vector. To do this we must first declare a variable and then read our sheet into it. Since we are talking about the eating habits of people we've taken our variable as food. Let's take a look at how to read our sheet into our variable food…

socialmedia=read.csv("socialmedia.csv", header = TRUE, stringsAsFactors = FALSE)

b) Importing Required Libraries:

Once we have successfully imported your dataset into the program we must now import all of the libraries that we will require in our program. We chose to make use of 2 very important libraries within our program. We require a library known as ***Rpart*** to create our ***Decision Tree.*** Another important library that we require known as ***e1071*** which will help us while we use ***Naïve Bayes Classification*** in our second part of our data analysis. Importing a package in **"R"** is a very simple process. We will simply use the following syntax….

*library(*Name of Package*)*

Let's take a further look into the packages that we will be using within our program.

library(Rpart)

Rpart.plot is a very important library that we use while we build our decision tree. This is due to fact that it help us create a clear and easy to understand tree. It properly aligns the texts and creates proper shapes and images for each node in the tree so that is neatly displayed and can be easily understood. It automatically scales the tree at hand and adjusts it to give it the best fit along with a suitable design. You will further understand its use as we go through the procedure step by step.

library(e1071)

This is an important library that we must include as it holds the ***Naïve Bayes Classifier*** within it. Without this library we can not analyze our data further to calculate which app most people felt most secure.

Now that we have taken a look into how to import the crucial libraries into our program let's take a look at how we use them.

C. Analysis & Presentation Of Our Data:

Now that we have imported all of the required data into our program we can now begin the process to begin its analysis. This step our process will not only contribute in studying the data but we will also look much deeper into how accurate as well as how volatile our data sets are.

a) Analysis of Our Survey:

i) Decision Tree:

Decision tree is a type of supervised learning algorithm that can be used in both regression and classification problems. It works for both categorical and continuous input and output variables. The Decision Tree uses a tree like graph to represent the data or decisions made. To simplify this it is basically a flowchart like structure where each internal node is responsible for testing a specific attribute. The children of each node represent the outcome of the test. the further down we go along

the tree the more analysis we are able to obtain from the node. The Root node represents the entire sample. When we take the outputs of the test to create the children for any given node we consider this act as **splitting.** The consecutive process of splitting each node from its root forms our Decision Tree. When we finally reach a stage where the roots can no longer be split into further nodes we call these nodes as the Terminal or the leave nodes.

There is one major problem that we will come across when we construct any Decision Tree. Sometimes our learning algorithm will begin to generate hypothesis that are inadequate the the dataset and in turn form many errors further down in the tree. Consider this as the phenomenon of pouring water into a glass full of water, The water wont show any effect on the glass but it's simply a waste of time as well as water. The same can occur in a Decision Tree. It's a waste to constantly test our dataset when we have already acquired our goal state as it will only further generate errors. However there are two methods we can follow to eliminate the process of overflowing in our Decision Tree. These are the the Pruning techniques. Let's take a closer look at these two methods.

Pre-pruning:

It is simply a process in which we stop the construction of the Decision Tree earlier then required. We will appoint a threshold value which will indicate how much data we want analysed by each node. If the node satisfies the given threshold limit then our algorithm will not further split the node. However it is very tough to determine the threshold for any set of data and decide when the tree should be stopped.

Post-pruning:

This is the process in which we will allow the tree to grow deeper and deeper until our algorithm identifies that overflow is occuring or till the tree is completed. If we are able to identify overflowing at any level of the tree then post pruning will come into action and will validate the entire tree from top to bottom to ensure the validity of our dataset. From doing this we will identify weather the further expansion of the node is necessary or not. If the further division of the node is not required we will simply declare the node at hand as the leaf or the terminal node.

ii) Construction of Decision Tree:

In order to construct our Decision tree we must first prioritize what attributes we would like to test from our dataset. to do this we will use the a module from the rpart library which will allow us to prioritize which attributes to test first. This is done as follows

Model<-
rpart(Chooseone~Accounts+Look+Access+Age+Post+

Gender,data=data,control=rpart.control(minsplit = 2))

In the model we have created we gave the first priority to the number of accounts that the user has registered. From here on out the dataset will be further classified based on Look, Access, Age, Post, Gender, and Data in the respective order. Minsplit is used to declare the number of children each parent can have. To view how our data has been assessed according to our priority we can enter the following command…

str(data)

The structure of data is:

On execution of this command we will get the following output

```
> str(data)
'data.frame':   544 obs. of  26 variables:
 $ Timestamp     : Factor w/ 542 levels "2019/02/15 10:00:26 PM GMT+5:30",..: 48
49 50 51 52 53 54 55 56 57 ...
 $ Name          : Factor w/ 201 levels "  K L UNIVERSITY",..: 68 23 23 95 8 73 8
6 198 90 86 ...
 $ Gender        : Factor w/ 2 levels "Female","Male": 1 1 1 1 1 2 1 2 1 2 ...
 $ Age           : Factor w/ 7 levels "15-20","21-30",..: 1 1 1 1 3 1 1 1 1 6 ...
 $ Used_platform : Factor w/ 58 levels "Facebook","Facebook;Instagram",..: 1 27 2
7 50 50 33 21 4 50 50 ...
 $ Accounts      : Factor w/ 8 levels "1","2","3","4",..: 5 2 2 4 4 4 3 5 3 8 ...
 $ Look          : Factor w/ 5 levels "10 + times","2-5 times a day",..: 1 3 3 3
1 1 2 1 2 1 ...
 $ Most_active   : Factor w/ 132 levels "02:00 am - 03:00 am",..: 111 111 124 111
62 114 62 111 117 9 ...
 $ Post          : Factor w/ 9 levels "Daily","For every 6 months",..: 4 2 2 6 2
3 6 9 8 5 ...
 $ Access        : Factor w/ 5 levels "any spare moment",..: 1 2 2 2 2 2 2 2 2 2
...
 $ College       : Factor w/ 4 levels "Break time","Class hours",..: 3 1 1 1 1 1
1 1 1 1 ...
 $ Workplace     : Factor w/ 5 levels "Leisures","Lunch time",..: 1 1 1 1 1 4 1 2
3 1 ...
 $ Night         : Factor w/ 2 levels "no","yes": 2 2 2 2 1 2 2 2 2 2 ...
 $ Morning       : Factor w/ 2 levels "no","yes": 2 2 2 1 2 2 1 2 1 2 ...
 $ Usage         : Factor w/ 182 levels "buying and selling",..: 94 132 132 20 11
1 14 171 111 111 96 ...
 $ Request       : Factor w/ 3 levels "No","Sometimes",..: 1 1 1 1 1 2 1 3 1 3 ..
.
 $ Secure_site   : Factor w/ 5 levels "Facebook","Hangouts",..: 5 5 5 5 5 5 5 3 5
5 ...
 $ Chooseone     : Factor w/ 6 levels "Facebook","Gmail",..: 1 6 6 6 6 3 5 3 6 6
...
 $ Visit         : Factor w/ 5 levels "Check out what's going on with my friends"
,..: 4 4 4 4 1 4 4 5 4 5 ...
 $ Devices       : Factor w/ 14 levels "Desktop","Desktop;Laptop",..: 12 12 12 12
12 12 10 12 12 9 ...
 $ Without_access: Factor w/ 4 levels "Fortnite","Half day",..: 3 4 4 4 2 3 4 3 1
2 ...
 $ Family_time   : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 2 1 2 ...
 $ Dual_Whatsapp : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
 $ Lock          : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 2 2 2 2 2 ...
 $ Share_lock    : Factor w/ 27 levels "Boy Friend","Boy Friend;None",..: 3 3 3 3
3 19 19 23 3 5 ...
 $ Rate          : num  4 3 3 1 3 1 3 4 2 5 ...
>
```

Fig. 1. (a)Structure of the data

From here we will enter the parameter for the tree that we would like to construct. We will enter the percentage which will will represent our threshold followed by the height of the tree that we would like to construct. The parameters can be set using the following syntax….

par(xpd=NA, mar=rep(0.7,4))

Once we have set the parameter for our tree we can plot our tree using the plot funtion.

plot(model,compress=T)

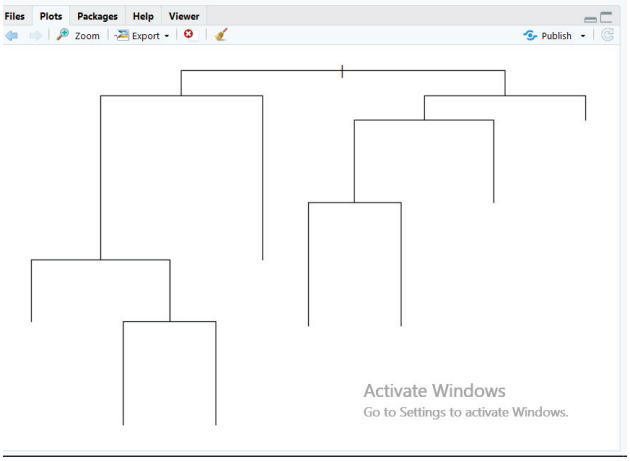

Fig. 1(b) Compressing the data in the form of tree

Once we have successfully platted our tree we will now want to add the text to each node within our tree. The text for each node is stored in our model while our tree has been successfully plotted. We will use the text function to establish this…

text(model,cex=0.7,use.n=T,fancy=F,all=T)

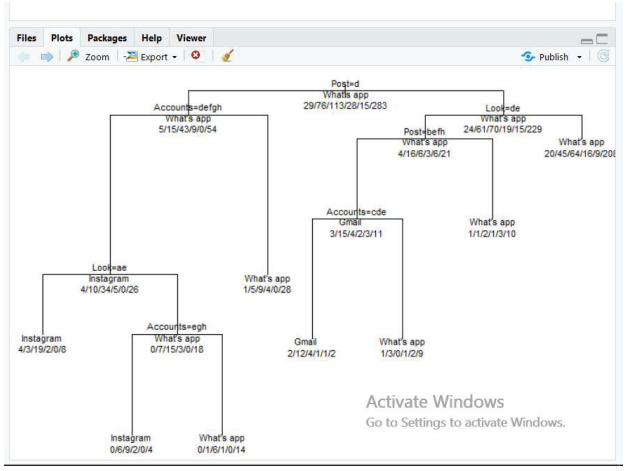

Fig. 1(c)Inserting the data in to the tree

We have successfully created our tree however it isn't quite neat nor is it organized. It is hard to identify on what basis each level on the tree is tested upon and how the splitting of each node has taken place. This is where the ***rpart.plot*** comes into role. As we have discussed about this important function before it is nothing more than simply the process of beautifying our tree. For you to get a better understanding observe how our tree changes after using this function.

We have successfully completed the construction of our Decision Tree however there is one thing we must check before we can make any conclusions. We must check whether the data that has been taken into account is complete or not. If we find cases in which data is missing from our analysis we will take responsibility to clear this using the following syntax to make sure that our data is complete….

is.na(model)

```
> par(xpd=NA,mar=rep(0.7,4))
> plot(model,compress=T)
> text(model,cex=0.7,use.n=T,fancy=F,all=T)
> rpart.plot(model)
> is.na(model)
             frame             where              call             terms           cptable
            method
             FALSE             FALSE             FALSE             FALSE             FALSE
              FALSE
             parms           control         functions          numresp            splits
            csplit
             FALSE             FALSE             FALSE             FALSE             FALSE
              FALSE
variable.importance                 y           ordered
             FALSE             FALSE             FALSE
>
```

Fig. 1 (d)Missing data

b) Identifying The Most Secure Social Platform :

In our second part of our project we tried to identify what social platform most people felt most secure upon. To do this we applied **Naïve Bayes Classification** to analyze our dataset. Before we go into the process of achieving this lets take a look into what exactly **Naïve Bayes Classification** is all about.

i) Naïve Bayes Classification:

Naïve Bayes is a simple, yet effective and commonly-used, machine learning classifier. **Naïve Bayes** classifiers are nothing more then set of algorithms that are based upon Bayes Theorems. It works on the principle stating that each and every feature is independent from the rest. One of the most common forms of using Naïve Bayes classifiers has been text classification. This method of classification is also an traditional solution for problems such as spam detection. Let's take a look at the algorithm that we will be using within our program….

$$P(A/B)= (P(B/A).P(A))/P(B)$$

ii) Implementation of Naïve Bayes Classification:

The first most important step in our process is to look upon the data we have at hand. Once we have successfully inputted our data into our program we will run the ***mydata*** function so that we can access our dataset. when we run this function it will provide the following metadata….

The next important step to perform upon our dataset is to provide it with a proper structure so that it a can be easily dealt with. This not only provides us with a structure but it also provides us with a neat and compact list as shown below…

str(data)

```
> str(mydata)
'data.frame':   544 obs. of  26 variables:
 $ Timestamp     : Factor w/ 542 levels "2019/02/15 10:00:26 PM GMT+5:30",..: 48 49 50 51 52 53 54
55 56 57 ...
 $ Name          : Factor w/ 201 levels "  K L UNIVERSITY",..: 68 23 23 95 8 73 86 198 90 86 ...
 $ Gender        : Factor w/ 2 levels "Female","Male": 1 1 1 1 1 2 1 2 1 2 ...
 $ Age           : Factor w/ 7 levels "15-20","21-30",..: 1 1 1 1 1 3 1 1 1 1 6 ...
 $ Used_platform : Factor w/ 58 levels "Facebook","Facebook;Instagram",..: 1 27 27 50 50 33 21 4 50
50 ...
 $ Accounts      : Factor w/ 8 levels "1","2","3","4",..: 5 2 2 4 4 4 3 5 3 8 ...
 $ Look          : Factor w/ 5 levels "10 + times","2-5 times a day",..: 1 3 3 3 1 1 2 1 2 1 ...
 $ Most_active   : Factor w/ 132 levels "02:00 am - 03:00 am",..: 111 111 124 111 62 114 62 111 117
9 ...
 $ Post          : Factor w/ 9 levels "Daily","For every 6 months",..: 4 2 2 6 2 3 6 9 8 5 ...
 $ Access        : Factor w/ 5 levels "any spare moment",..: 1 2 2 2 2 2 2 2 2 2 ...
 $ College       : Factor w/ 4 levels "Break time","Class hours",..: 3 1 1 1 1 1 1 1 1 1 ...
 $ Workplace     : Factor w/ 5 levels "Leisures","Lunch time",..: 1 1 1 1 1 4 1 2 3 1 ...
 $ Night         : Factor w/ 2 levels "no","yes": 2 2 2 2 1 2 2 2 2 2 ...
 $ Morning       : Factor w/ 2 levels "no","yes": 2 2 2 1 2 2 1 2 1 2 ...
 $ Usage         : Factor w/ 182 levels "buying and selling",..: 94 132 132 20 111 14 171 111 111 9
6 ...
 $ Request       : Factor w/ 3 levels "No","Sometimes",..: 1 1 1 1 1 2 1 3 1 3 ...
 $ Rate          : Factor w/ 7 levels "01-May","02-May",..: 5 3 4 1 4 1 4 6 2 7 ...
 $ Secure_site   : Factor w/ 5 levels "Facebook","Hangouts",..: 5 5 5 5 5 5 5 3 5 5 ...
 $ Chooseone     : Factor w/ 6 levels "Facebook","Gmail",..: 1 6 6 6 6 3 5 3 6 6 ...
 $ Visit         : Factor w/ 5 levels "Check out what's going on with my friends",..: 4 4 4 4 1 4 4
5 4 5 ...
 $ Devices       : Factor w/ 14 levels "Desktop","Desktop;Laptop",..: 12 12 12 12 12 12 10 12 12 9
...
 $ Without_access: Factor w/ 4 levels "Fortnite","Half day",..: 3 4 4 4 2 3 4 3 1 2 ...
 $ Family_time   : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 2 1 2 ...
 $ Dual_Whatsapp : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
 $ Lock          : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 2 2 2 2 2 ...
 $ Share_lock    : Factor w/ 27 levels "Boy Friend","Boy Friend;None",..: 3 3 3 3 3 19 19 23 3 5 ..
>
```

Fig. 2 (a)Structure of the data

for testing. We will now store the split data into the mtraining and mtesting variables separately ….

mtraining<-mydata[dex,]

mtesting<-mydata[-dex,]

Data	10 obs. of 5 variables
data	544 obs. of 26 variables
df	10 obs. of 4 variables
model	List of 15
mtesting	164 obs. of 26 variables
mtraining	380 obs. of 26 variables
mydata	544 obs. of 26 variables
NB	List of 4

Fig. 2 (b)Testing and training data

From here we will import Library e1071 as it is the library that contains all the algorithms and functions that are required for **Naïve Bayes Classification….**

library(e1071)

We will now take our training dataset and apply **Naïve Bayes** theorems on our datasets with the following commands in **"R"**.

NB<-naiveBayes(Secure-site ~.,data=mtraining)

From the data that we have successfully been able to analyze we will now make predictions upon which social media platform was felt most secure by its users.

predNB1<-predict(NB,mtesting,type=c("class"))

We will take all the data that we have analyzed till now and simply print it out in a tabular format and then use this to form a bar graph so that it is far easier to understand….

table(mtesting$Secure_site,predNB1)

```
> predNB1<-predict(NB,mtesting,type=c("class"))
> table(mtesting$Secure_site,predNB1)
           predNB1
            Facebook Gmail Instagram Telegram Tiktok What's app
  Facebook         0     0         1        1      0          2
  Hangouts         0     8         5        0      1          5
  Instagram        1     1        16        0      0         10
  Telegram         0     6         2        0      0         10
  Whats app        2     8        11        0      0         74
>
```

Fig..2 (c)Tabular form of testing data

plot(predNB1)

D. Result:

The output for the tested data

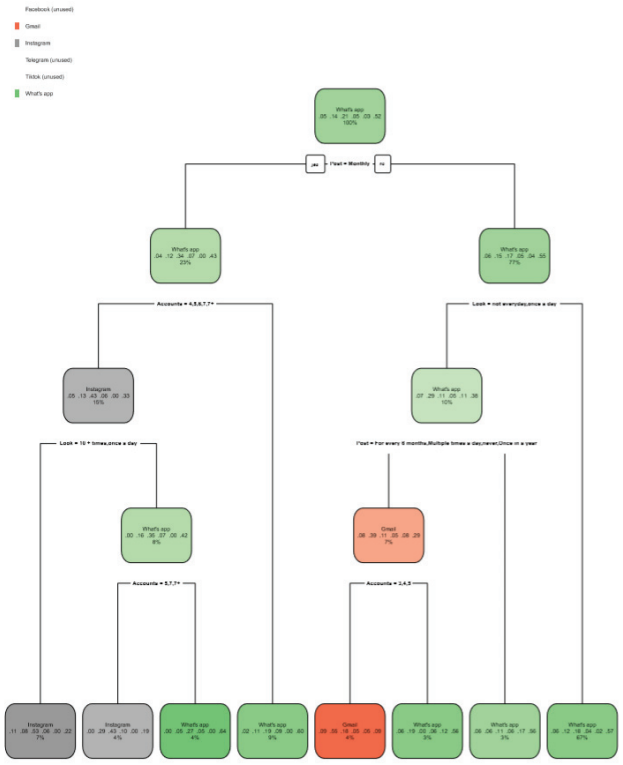

Fig. 3. (a) Decision tree for given data

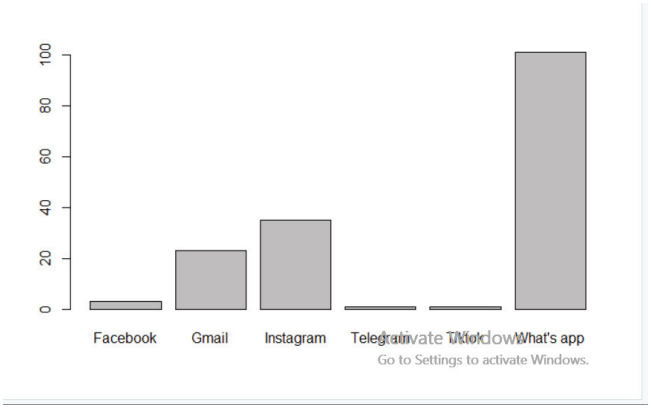

Fig. 3. (b) Classifying the given data using Naïve Bayes

III CONCLUSION

In this social media survey we have used two algorithms they were decision tree algorithm and Naïve Bayes classification.

In the Decision tree algorithm we have predicted which social media site is more liked by the people and do they feel it is secure. According to that we can predict which age group are more addicted to social media.

In the Naïve Bayes classification we have predicted Which social media site is more liked by the people. In the survey we asked the people to select one of the social media site so that we can predict which of the social media app is used most and which of the social media site do they like more among all other social media sites.

REFERENCES

[1] F. Amato, A. Castiglione, A. D. Santo, V. Moscato, A. Pi-cariello, F. Persia, G. Sperli, "Recognizing human behaviours in online social networks", *Computers Security pages* -, 2017.

[2] Kagan, V., & Subrahmanian, V. S. (2018). Understanding Multi-Stage, Multi-Modal, Multimedia Events in Social Media. 2018 International Workshop on Social Sensing (SocialSens).

[3] Chumwatana, T., & Chuaychoo, I. (2017). Using social media listening technique for monitoring people's mentions from social media: A case study of Thai airline industry. 2017 2nd International Conference on Knowledge Engineering and Applications (ICKEA).

[4] A Survey on Privacy and Security in Online Social Networks Imrul Kayes, University of South Florida Adriana Iamnitchi, University of South Florida.

[5] Bhargava, Geetha. (2018). Sentimental analysis on social media data using R programming. INTERNATIONAL JOURNAL OF ENGINEERING AND TECHNOLOGY (UAE). 7. 10.14419/ijet.v7i2.31.13402.

[6] Wang, Z., Chong, C. S., Lan, L., Yang, Y., Beng Ho, S., & Tong, J. C. (2016). Fine-grained sentiment analysis of social media with emotion sensing. 2016 Future Technologies Conference (FTC).