



Lagdu Singh Charitable Trust's (Regd.)

THAKUR COLLEGE OF ENGINEERING & TECHNOLOGY

Autonomous College Affiliated to University of Mumbai

Approved by All India Council for Technical Education (AICTE) and Government of Maharashtra (GoM)

Conferred Autonomous Status by University Grants Commission (UGC) for 10 years w.e.f. A.Y 2019-20

Amongst Top 200 Colleges in the Country, Ranked 193rd in NIRF India Ranking 2019 in Engineering College category

• ISO 9001:2015 Certified • Programmes Accredited by National Board of Accreditation (NBA), New Delhi

• Institute Accredited by National Assessment and Accreditation Council (NAAC), Bangalore

A.Y. 2020 – 21 Institutional Internship

Data Science using R Programming

Project Report on “WhatsApp Chat Analysis”

Team Members Details:

Kiran Maharana

Surajit Mondal

Gayatri Menon

Instructor:

Dr. Anand Khandre

**Thakur College of Engineering and Technology, Kandivali
2021-22**

INDEX

CHAPTER	NAME OF TOPIC	PAGE NO.
1.	ABSTRACT	3
2.	INTRODUCTION	3
3.	PROBLEM STATEMENT	4
4.	FLOWCHART	4
5.	TECHNOLOGIES USED	5
6.	IMPLEMENTATION	8
7.	RESULTS & DISCUSSION	10
8.	CONCLUSION	13
9.	FUTURE SCOPE	13
10.	REFERENCES	14

FIGURE INDEX

FIGURE NO.	FIGURE DETAILS	PAGE NO.
1.	FLOWCHART	5
2.	DATASET	3
3.	OUTPUT	24 - 44

ABSTRACT:

WhatsApp applications has recently emerged as a substitute for SMS in developing countries. It includes variety of functions such as sharing live location, files, video, image, audio and text messages. It generated huge volume of data which has not yet been thoroughly researched. What does other person feel about us is the most frequently self-asked question, everyone has the curiosity of what other person thinks about the other while having a conversation, judging the other person can't be done perfectly. We present an extensive study of the usage of the WhatsApp social network and presenting the analysis of over 40k text and media messages. We have tried to implement how we can extract keywords from text messages and target ads to the users. Lastly, analysing what are factors and terms influencing user to click on ads shown to them. We present a detailed discussion about the every specific terms needed to analyse from a WhatsApp chat.

KEYWORD:

WhatsApp, Chats, Sentiment Analysis, Target Ads, Clicked-Ad Prediction, Naïve Bayes, Decision Tree

INTROUCTION:

The most used and efficient method of communication in recent times is an application called WhatsApp. WhatsApp is playing a vital role in messaging service. It comes up with many innovative ideas for some it has seen set-backs too. WhatsApp chats consists of various kinds of conversations held among group of people. This chat consists of various topics. This information can provide lots of data. WhatsApp claims that nearly 55 billion messages are sent each day. The average user spends 195 minutes per week on WhatsApp, and is a member of plenty of groups. By providing service in over 100 countries and having active users over 2.5 billion.

Sentiment analysis is a method or a subprocess of natural language processing and data mining that is basically used for the identification of user's opinion or emotion. Also, this concept works on the core concepts of machine learning where a data set is used to train and process that data and a model is generated and used for evaluation of emotion of test chat. This process has evolved into a very powerful field in text analysis. Every day, we generate a huge amount of text online but analysing this text data isn't an easy task. Converting text into structured information to analyse with a machine will be a complex task. In recent years, Text mining has become a lot more accessible for data analysts, developers and data scientists.

WhatsApp chat message and plot a pie chart visualizing the percentage of texts that have a Positive, Negative, or Neutral connotation or Sentiment to it. So here we have learned how to analyse WhatsApp Chats using R

So, in this project we will be covering four parts – firstly, we will analyse the chat messages of the user and then we learn about their sentiment, feeling and emotion over a period of time, following we will learn about what are the thing a user is talking in the group and we try to suggest which type of ads can be target to them and lastly, we analyse what are the factors affecting and influencing user to click on a advertisement.

PROBLEM STATMENT:

WhatsApp seems to become increasingly important not just as a messaging service but also as a social network in which Group Chat capabilities plays a vital role.

So, understanding and analysing of text messages, sentiments and emotions can give us precious insights. Businesses can know feelings of their potential customers about a product or a service to target ads, know click-ad success rate and factors affecting it and learn more about them.

MOTIVATION:

If you have ever emailed a WhatsApp chat to yourself, (or someone else, if that's how you roll) you may have noticed that it includes a handful of details that can be used to analyze a group of texts as well as the entire chat history. WhatsApp allows you to export/share your chat history with contact as a .txt file and it has the following structure:

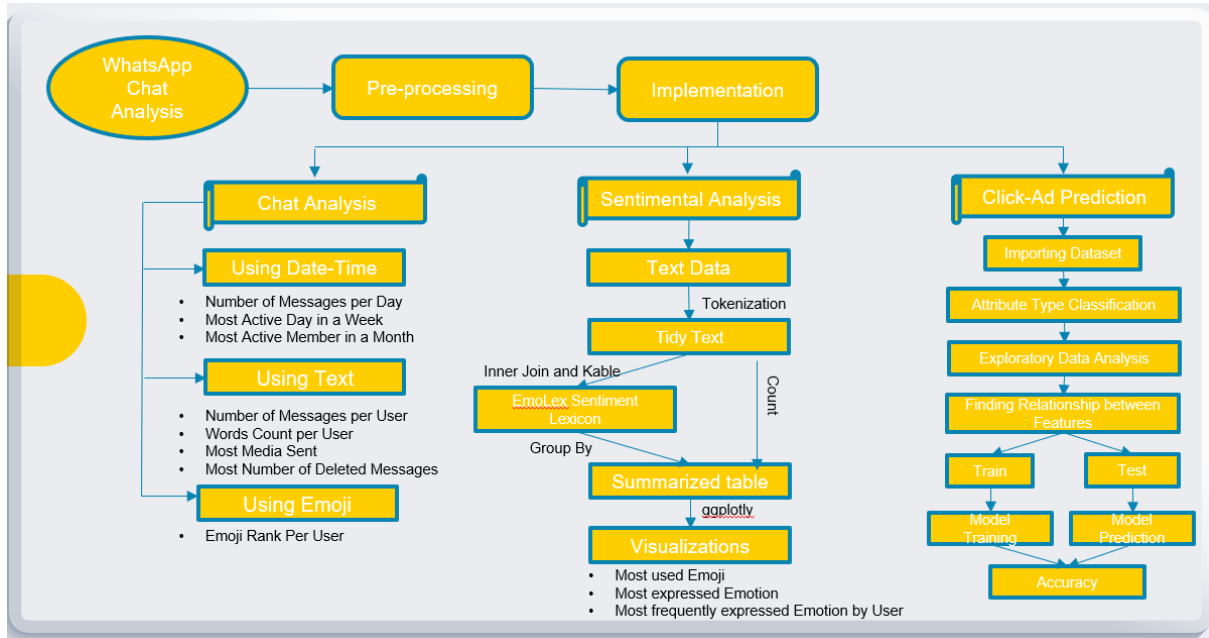
We are going to take advantage of this data, so let's take a look at various mining patterns from Chats. The use of instant messaging platforms such as WhatsApp for civic and political purposes has been observed and reported to be growing faster than other social media platforms especially in recent years. Using empirical research on WhatsApp studies published from 2009 to 2019 as its corpus of data, this article systematically reviews them to provide more robust conclusions about WhatsApp and its relationship with political and/or civic engagement.

We wanted to seeks answers of the three central questions related to WhatsApp and engagement:



- 1) What are the motivations in using WhatsApp and how do they manifest in the use of WhatsApp as a communication tool?
- 2) What is the role of WhatsApp in industrial engagement?
- 3) How do researchers study the use of WhatsApp in order to understand human nature?

FLOWCHART:

The flow chart represent the flow of the project which consists of pre-processing and implementation of chat analysis, sentimental analysis and click-ad prediction.



TECHNOLOGIES USED:

-  RStudio
-  R - GUI
-  R Programming

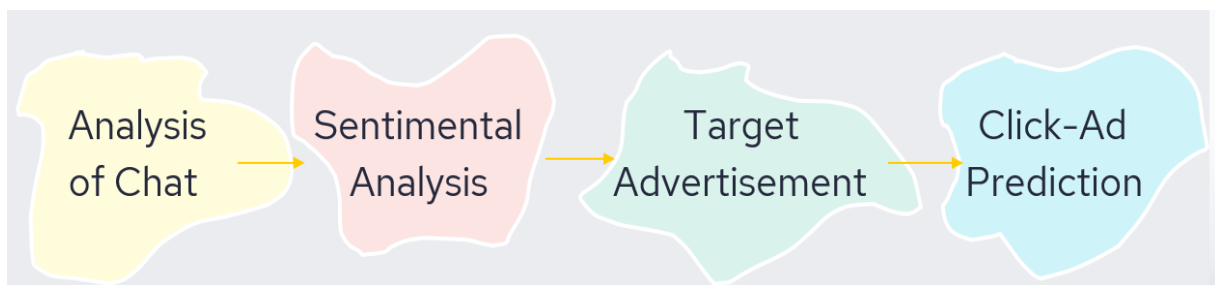
IMPLEMENTATION:

Packages Used –

- | | |
|------------------------|------------------------|
| ❖ library("rwhatsapp") | ❖ library("rlang") |
| ❖ library("dplyr") | ❖ library("tidyr") |
| ❖ library("ggplot2") | ❖ library("tidyverse") |
| ❖ library("lubridate") | ❖ library("ggimage") |
| ❖ library("ggplot2") | ❖ library("tidytext") |
| ❖ library("plotly") | ❖ library("psych") |

- ❖ library("naivebayes")
- ❖ library("kableExtra")
- ❖ library("naniar")
- ❖ library("caret")
- ❖ library("party")

About:



Analysis of Chat:

Using Data-Time we will be analysing number of messages per day, most active day in a week, most active member in a month.

Using Text we will be analyzing number of messages per user, words count per user, most media sent, most number of deleted messages.

Using Emoji we will be analysing most sent emoji by user.

Sentimental Analysis:

To understand the feeling of the user during chatting in the group. We will be analysing sentiment of the user.

Here, we will be using EmoLex technique. NRC lexicon categorizes the words in binary fashion into categories of positive, negative, anger, anticipation, disgust, fear, joy, sadness, surprise and trust.

Target Advertisement:

Using chat we will be analysing about what stuff he is talking about. We will be using grep() function here. Grep function return the indices of vector elements that contains the character.

Click-Ad Prediction:

To understand what are factors affecting and influencing user to click on the ads and how we can make this process easier and more feasible. We will be implementing model.

We will be using naïve bayes and decision tree and try to find best method.

Dataset:

1	2020-06-29 06:10:54	Tejas	congrats darsh, aarav, utkarsh & rushal 🎉🎉🎉🎉	4	c["<U+0001F973>","<U+0001F973>","<U+0001F973>","...	c["partyng face","partyng face","partyng face","partyng f...	Mon	Jun	29
2	2020-06-29 06:10:54	Aarav	thanks	5	NULL	NULL	Mon	Jun	29
3	2020-06-29 06:11:54	Tejas	ab party to banti he 🎉🎉	6	c["<U+0001F973>","<U+0001F929>"]	c["partyng face","star-struck"]	Mon	Jun	29
4	2020-06-29 06:11:54	Darsh	thank you	7	NULL	NULL	Mon	Jun	29
5	2020-06-29 06:11:54	Aarav	👍	8	👍	thumbs up	Mon	Jun	29
6	2020-06-29 07:40:54	Utkarsh	👍	9	👍	thumbs up: light skin tone	Mon	Jun	29
7	2020-06-29 07:40:54	Utkarsh	🇮🇳🇮🇳	10	c["<U+0001F1EE> <U+0001F1F3>","<U+0001F1EE> <U+00...	c["flag: India","flag: India"]	Mon	Jun	29
8	2020-06-29 07:47:54	Utkarsh	congratulations to all! we all have completed our first year e...	11	c["<U+0001F973>","<U+0001F973>"]	c["partyng face","partyng face"]	Mon	Jun	29
9	2020-06-29 07:47:54	Utkarsh	all the best for second year!	12	NULL	NULL	Mon	Jun	29
10	2020-06-29 07:47:54	Darsh	<media omitted>	13	NULL	NULL	Mon	Jun	29
11	2020-06-29 07:47:54	Utkarsh	<media omitted>	14	NULL	NULL	Mon	Jun	29
12	2020-06-29 08:44:54	Tejas	59 chinese app banned 🎉🎉	15	c["<U+0001F973>","<U+0001F973>"]	c["partyng face","partyng face"]	Mon	Jun	29
13	2020-06-29 08:44:54	Darsh	kaha aya?	16	NULL	NULL	Mon	Jun	29
14	2020-06-29 08:44:54	Tejas	zee news	17	NULL	NULL	Mon	Jun	29
15	2020-06-29 08:44:54	Tejas	you deleted this message	18	NULL	NULL	Mon	Jun	29
16	2020-06-29 08:45:54	Darsh	<media omitted>	19	NULL	NULL	Mon	Jun	29
17	2020-06-29 08:45:54	Darsh	<media omitted>	20	NULL	NULL	Mon	Jun	29
18	2020-06-29 08:45:54	Darsh	<media omitted>	21	NULL	NULL	Mon	Jun	29
19	2020-06-29 08:45:54	Darsh	<media omitted>	22	NULL	NULL	Mon	Jun	29
20	2020-06-29 08:45:54	Darsh	<media omitted>	23	NULL	NULL	Mon	Jun	29
21	2020-06-29 08:45:54	Darsh	<media omitted>	24	NULL	NULL	Mon	Jun	29
22	2020-06-29 08:45:54	Darsh	maza agaya	25	NULL	NULL	Mon	Jun	29
23	2020-06-29 08:46:54	Tejas	you deleted this message	26	NULL	NULL	Mon	Jun	29
24	2020-06-29 08:46:54	Tejas	you deleted this message	27	NULL	NULL	Mon	Jun	29
25	2020-06-29 08:51:54	Darsh	khatam abhi	28	NULL	NULL	Mon	Jun	29
26	2020-06-29 08:51:54	Tejas	pubg nahi he list me 🎉	29	🎉	slightly frowning face	Mon	Jun	29
27	2020-06-29 08:52:54	Darsh	hai	30	NULL	NULL	Mon	Jun	29
28	2020-06-29 08:52:54	Tejas	no	31	NULL	NULL	Mon	Jun	29
29	2020-06-29 08:52:54	Darsh	ok	32	NULL	NULL	Mon	Jun	29
30	2020-06-29 08:53:54	Tejas	<media omitted>	33	NULL	NULL	Mon	Jun	29
31	2020-06-29 08:53:54	Darsh	mazza ageya	34	NULL	NULL	Mon	Jun	29
32	2020-06-29 09:32:54	Utkarsh	<media omitted>	35	NULL	NULL	Mon	Jun	29
33	2020-06-29 10:55:54	Sandhya	abhi tak first sem ka darsh diya hi nhi h to ab	36	NULL	NULL	Mon	Jun	29
34	2020-06-29 10:56:54	Sandhya	to ab ye sem vala kya milega 🎉	37	🎉	rolling on the floor laughing	Mon	Jun	29

1	68.95	35	61833.90	256.09	Cloned 5th generation orchestration	Wrightburgh	0	Tunisia	2016-03-27 00:53:11	0
2	80.23	31	68441.85	193.77	Monitored national standardization	West Jodi	1	Nauru	2016-04-04 01:39:02	0
3	69.47	26	59785.94	236.50	Organic bottom-line service-desk	Davidton	0	San Marino	2016-03-13 20:35:42	0
4	74.15	29	54806.18	245.89	Triple-buffered reciprocal time-frame	West Terrifurt	1	Italy	2016-01-10 02:31:19	0
5	68.37	35	73899.99	225.58	Robust logistical utilization	South Manuel	0	Iceland	2016-06-03 03:36:18	0
6	59.99	23	59761.56	226.74	Sharable client-driven software	Jamieberg	1	Norway	2016-05-19 14:30:17	0
7	88.91	33	53852.85	208.36	Enhanced dedicated support	Brandonstad	0	Myanmar	2016-01-28 20:59:32	0
8	66.00	48	24593.33	131.76	Reactive local challenge	Port Jefferybury	1	Australia	2016-03-07 01:40:15	1
9	74.53	30	68862.00	221.51	Configurable coherent function	West Colin	1	Grenada	2016-04-18 09:33:42	0
10	69.88	20	55642.32	183.82	Mandatory homogeneous architecture	Ramirezton	1	Ghana	2016-07-11 01:42:51	0
11	47.64	49	45632.51	122.02	Centralized neutral neural-net	West Brandonton	0	Qatar	2016-03-16 20:19:01	1
12	83.07	37	62491.01	230.87	Team-oriented grid-enabled Local Area Network	East Theresashire	1	Burundi	2016-05-08 08:10:10	0
13	69.57	48	51636.92	113.12	Centralized content-based focus group	West Katiefurt	1	Egypt	2016-06-03 01:14:41	1
14	79.52	24	51739.63	214.23	Synergistic fresh-thinking array	North Tara	0	Bosnia and Herzegovina	2016-04-20 21:49:22	0
15	42.95	33	30976.00	143.56	Grass-roots coherent extranet	West William	0	Barbados	2016-03-24 09:31:49	1
16	63.45	23	52182.23	140.64	Persistent demand-driven interface	New Travistown	1	Spain	2016-03-09 03:41:30	1
17	55.39	37	23936.86	129.41	Customizable multi-tasking website	West Dylanberg	0	Palestinian Territory	2016-01-30 19:20:41	1
18	82.03	41	71511.08	187.53	Intuitive dynamic attitude	Pruittmouth	0	Afghanistan	2016-05-02 07:00:58	0
19	54.70	36	31087.54	118.39	Grass-roots solution-oriented conglomeration	Jessicastad	1	British Indian Ocean Territory (Chagos Archipelago)	2016-02-13 07:53:55	1
20	74.58	40	23821.72	135.51	Advanced 24/7 productivity	Millertown	1	Russian Federation	2016-02-27 04:43:07	1
21	77.22	30	64802.33	224.44	Object-based reciprocal knowledgebase	Port Jacqueline	1	Cameroon	2016-01-05 07:52:48	0
22	64.59	35	60015.57	226.54	Streamlined non-volatile analyzer	Lake Nicole	1	Cameroon	2016-03-18 13:22:35	0
23	41.49	52	32635.70	164.83	Mandatory disintermediate utilization	South John	0	Burundi	2016-05-20 08:49:33	1
24	87.29	36	61628.72	209.93	Future-proofed methodical protocol	Pamelamouth	1	Korea	2016-03-23 09:43:43	0
25	41.39	41	68962.32	167.22	Exclusive neutral parallelism	Harperborough	0	Tokelau	2016-06-13 17:27:09	1
26	78.74	28	64828.00	204.79	Public-key foreground groupware	Port Danielleberg	1	Monaco	2016-05-27 15:25:52	0
27	48.53	28	38067.08	134.14	Ameliorated client-driven forecast	West Jeremyside	1	Tuvalu	2016-02-08 10:46:14	1
28	51.95	52	58295.82	129.23	Monitored systematic hierarchy	South Cathylfurt	0	Greece	2016-07-19 08:32:10	1
29	70.20	34	32708.94	119.20	Open-architected impactful productivity	Palmeriside	0	British Virgin Islands	2016-04-14 05:08:35	1
30	76.02	22	46179.97	209.82	Business-focused value-added definition	West Guybury	0	Bouvet Island (Bouvetøya)	2016-01-27 12:38:16	0
31	67.64	35	51473.28	267.01	Programmable asymmetric data-warehouse	Phelpschester	1	Peru	2016-07-02 20:23:15	0
32	86.41	28	45593.93	207.48	Digitized static capability	Lake Melindamouth	1	Aruba	2016-03-01 22:13:37	0
33	59.05	57	25583.29	169.23	Digitized global capability	North Richardburgh	1	Maldives	2016-07-15 05:05:14	1
34	55.60	23	30227.98	212.58	Multi-layered 4th generation knowledge user	Port Cassie	0	Senegal	2016-01-14 14:00:09	1

Pre-processing:

In Dataset in WhatsApp Chat -

The data with null value are required to be clean/removed from the dataset. So, the message with no author or author as NA is removed and source column is dropped.

In Dataset in Advertisement Click Prediction –

The columns which are required are being dropped from the csv file Ad.Topic.Line, City, Country, Timestamp.

Model Used:

1. Naive Bayes classification

Naive Bayes is a Supervised Non-linear classification algorithm in R Programming. Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Baye's theorem with strong(Naive) independence assumptions between the features or variables. The Naive Bayes algorithm is called “Naive” because it makes the assumption that the occurrence of a certain feature is independent of the occurrence of other features

Theory -

Naive Bayes algorithm is based on Bayes theorem. Bayes theorem gives the conditional probability of an event A given another event B has occurred.

$$P(A|B) = \frac{P(B|A) P(A)}{P(B)}$$

where,

$P(A|B)$ = Conditional probability of A given B.

$P(B|A)$ = Conditional probability of B given A.

$P(A)$ = Probability of event A.

$P(B)$ = Probability of event B.

2. Decision Tree algorithm

where,

$P(A|B)$ = Conditional probability of A given B.

$P(B|A)$ = Conditional probability of B given A.

$P(A)$ = Probability of event A.

$P(B)$ = Probability of event B.

Working of a Decision Tree -

- **Partitioning:**

It refers to the process of splitting the data set into subsets. The decision of making strategic splits greatly affects the accuracy of the tree. Many algorithms are used by the tree to split a node into sub-nodes which results in an overall increase in the clarity of the node with respect to the target variable. Various Algorithms like the chi-square and Gini index are used for this purpose and the algorithm with the best efficiency is chosen.

- **Pruning:**

This refers to the process wherein the branch nodes are turned into leaf nodes which results in the shortening of the branches of the tree. The essence behind this idea is that overfitting is avoided by simpler trees as most complex classification trees may fit the training data well but do an underwhelming job in classifying new values.

- **Selection of the tree:**

The main goal of this process is to select the smallest tree that fits the data due to the reasons discussed in the pruning section.

Implemented Models:

Naive bayes -

Naïve Bayes classification is a kind of simple probabilistic classification methods based on Bayes' theorem with the assumption of independence between features. The model is trained on training dataset to make predictions by predict() function

Decision Tree -

Decision tree is a graph to represent choices and their results in form of a tree. The nodes in the graph represent an event or choice and the edges of the graph represent the decision rules or conditions. It is mostly used in Machine Learning and Data Mining applications using R.

Functions used:

1. pairs.panels()
2. vis_miss()
3. naive_bayes()
4. Ctree()
5. Predict()
6. Cbind()
7. Confusionmatrix()

Performance Evaluation

Naive Bayes classification:

Misclassification: 25%

Accuracy: 75%

Decision Tree:

Misclassification: 30%

Accuracy: 70%

Comparing Various Models

Naïve Bayes VS Decision Tree:

We conclude that Naive Bayes Algorithm gives us the maximum accuracy for determining the click probability.

Why Naive Bayes is better ?

A Naive Bayes classifier will converge quicker than discriminative models like logistic regression, so you need less training data. Since we have smaller dataset, that is the reason we have got higher accuracy for Naive Bayes.

Code:

```
#####  
#####  
##### Group No.: 13 #####  
##### Group Members: Surajit Mondal, Kiran Maharana, Gayatri Menon #####  
##### Internship Track: Data Science Using R #####
```

```
##### Incharge: Dr. Anand Khandare #####
##### Project Name: WhatsApp Chat Analysis #####
#####
#####

# Loading Libraries

library("rwhatsapp")
library("dplyr")
library("ggplot2"); theme_set(theme_minimal())
library("lubridate")
library("ggplot2")
library("plotly")
library("rlang")
library("tidyr")
library("tidyverse")
library("ggimage")
library("tidytext")

# -----
# READING THE DATASET
# -----

chat <- rwa_read("C:/Users/Kiran Maharana/Documents/SEM IV/Summer Internship/d
ataset.txt")

# Understanding the structure of our Dataset

# Display of Head and Tail of Dataset
head(chat)
tail(chat)

# Display the Datatypes of Columns of Dataset
sapply(chat, class)

# Display the Number of Columns and Rows in Dataset
ncol(chat)
nrow(chat)

# -----
```

```
# Data Pre-processing
# -----

# Remove Messages without Author
chat = chat %>% filter(!is.na(author))

# Dropping column source which is insignificant for our analysis
chat = subset(chat, select = -c(source) )

# -----
# ANALYSIS OF WHATSAPP CHAT
# -----

# ----- Display the messages per day -----

message_day <- chat %>%
  mutate(day = date(time)) %>%
  count(day) %>%
  ggplot(aes(x = day, y = n)) +
  theme_bw()+
  geom_bar(stat = "identity",fill = "#0c4c8a") +
  ylab("No. of Messages") + xlab("Days") +
  ggtitle("Messages per Day")

ggplotly(message_day)

# ----- Display the Number of Messages of each User -----

ggplotly(
  chat %>%
    count(author) %>%
    ggplot(aes(x = reorder(author, n), y = n ,fill = author)) +
    geom_bar(stat = "identity") +
    ylab("Totals") + xlab("Group Members") +
    coord_flip() +
    ggtitle("Number of messages sent") +
    theme_minimal()
)
```

```
# ----- Emojis Rank per User -----
--

plotEmojis <- chat %>%
  unnest(c(emoji, emoji_name)) %>%
  mutate( emoji = str_sub(emoji, end = 1)) %>%
  count(author, emoji, emoji_name, sort = TRUE) %>%
  # Plot top 8 Emojis per User
  group_by(author) %>%
  top_n(n = 8, n) %>%
  slice(1:8) %>%

  # Create an Image URL with the Emoji UNICODE
  mutate( emoji_url = map_chr(emoji,
                                ~paste0('https://abs.twimg.com/emoji/v2/72x72/',
as.hexmode(utf8ToInt(.x)),'.png')) )

# Plot Data
plotEmojis %>%
  ggplot(aes(x = reorder(emoji, -n), y = n)) +
  geom_col(aes(fill = author, group=author), show.legend = FALSE, width = .20)
+
  # Use to fetch an Emoji Ping Image https://abs.twimg.com
  geom_image(aes(image=emoji_url), size=.13) +
  ylab('') +
  xlab('') +
  facet_wrap(~author, ncol = 5, scales = 'free') +
  ggtitle('Most used emojis by users') +
  theme_minimal() +
  theme(axis.text.x = element_blank())

# ----- Word Count per User -----
--

library("stopwords")

to_remove <- c(stopwords(language = "en"),
               "media", "omitted", "ref", "dass", "schon", "mal", "android.s.w
t",
               "this", "message", "deleted","ka",'bhi','ye','nono','1','2','3'
,
```

```
'4','5','surajit','aditya','mondal','karan','h','nai','toh','hi',
',
'hua','na','j','int','prashant','chaurasia','jha','vinay')

word_count <- chat %>%
  unnest_tokens(input = text,
                output = word) %>%
  filter(!word %in% to_remove) %>%
  count(author, word, sort = TRUE) %>%

# Top 5 words used by the User
group_by(author) %>%
top_n(n = 5, n) %>%
slice(1:20) %>%
ungroup() %>%
arrange(author, desc(n)) %>%
mutate(order=row_number()) %>%
ggplot(aes(x = reorder(word, n), y = n, fill = author, color = author)) +
  geom_col(show.legend = FALSE, width = .1) +
  geom_point(show.legend = FALSE, size = 3) +
  ylab('Number of Occurrence') +
  xlab('Words Used') +
  coord_flip() +
  facet_wrap(~author, ncol = 3, scales = 'free') +
  ggtitle('Most Words used by Users') +
  theme_minimal()

ggplotly(word_count)

# ----- Most Active Day in a Week -----
--

chat <- chat %>% mutate(Dow = wday(as.Date(chat$time), label=TRUE))
dow <- chat %>% filter(Dow != '') %>% group_by(Dow) %>% summarise(count = n())

active_day <- ggplot(dow,aes(x=Dow,y = count, fill = Dow))+
  geom_bar(stat = "identity")+
  xlab("Days of the week")+
  ylab("Messages")+
  coord_flip()+
  geom_text(aes(label = scales::comma(count)), hjust = 3) +
  ggtitle("Days most active")+
  theme_minimal()
```



```
ggplotly(active_day)

# ----- Most Active Member Each Month -----
--

chat <- chat %>% mutate(months = month(as.POSIXct(chat$time, '%m'), label = TRUE))
mnths <- chat %>% filter(months != '') %>% group_by(months) %>% summarise(mcount = n())
actMember <- chat %>% filter(months != '') %>% group_by(months, author) %>% summarise(scount = n()) %>% slice(which.max(scount))
mnthsactMember <- merge(mnths, actMember, by="months")

ggplot(mnthsactMember)+
  geom_bar(aes(x=months, y = mcount, fill = months), stat = "identity", width = 1)+
  geom_point(aes(x=months, y = scount, color = author),
    size = 4, alpha = 0.5,
    stat = "identity",
  )+
  # geom_text(aes(x=months, y = scount, label = Name), vjust = 0.5, hjust = -1, color = "white")+
  geom_label(aes(x=months, y = scount, label = paste0(author, " (", scount, ")")),
    fill = 'black', vjust = 0.5, hjust = -0.4, color = "white", alpha = 0.5, size = 3.5
  )+
  xlab("Months")+
  ylab("Messages")+
  coord_flip()+
  facet_wrap(~author, ncol = 2, scales = "free_y") +
  ggtitle("Most Active Member Each Month")+
  theme_minimal(base_size = 10)

# ----- Most Media Sent by User -----
--

media_count = chat %>%
  group_by(author) %>%
  filter(text=="<Media omitted>") %>%
  summarise(count_of_media=n()) %>%
  arrange(desc(count_of_media))
```

```
media_count_1=media_count[1:7,]

p <- media_count_1 %>%
  ggplot(aes(x = reorder(author,count_of_media), y = count_of_media,fill=author)) +
  theme_bw()+
  geom_bar(stat = "identity") +
  ylab("Media count") + xlab("Users") +
  ggtitle("Most media sent")

ggplotly(p)

# ----- Member who Deleted most number of Messages -----
--

deleted_messages_count = chat %>%
  group_by(author) %>%
  filter(text=="This message was deleted") %>%
  summarise(count_of_deleted_message=n()) %>%
  arrange(desc(count_of_deleted_message))

deleted_messages_count_1 = deleted_messages_count[1:7,]

p <- deleted_messages_count_1 %>%
  ggplot(aes(x = reorder(author,count_of_deleted_message), y = count_of_deleted_message,fill=author)) +
  theme_bw()+
  geom_bar(stat = "identity") +
  ylab("count of messages deleted") + xlab("Users") +
  ggtitle("Most number of Deleted Messages by the user")

ggplotly(p)

# -----
--
# SENTIMENTAL ANALYSIS OF CHATS USING EMOJIS USED
# BY USING NRC WORD-EMOTION LEXICON (EmoLex)
# -----
--

# Library for Emoji PNG Image Fetch from https://abs.twimg.com
```

```
# Emoji Ranking
plotEmojis <- chat %>%
  unnest(emoji, emoji_name) %>%
  mutate( emoji = str_sub(emoji, end = 1)) %>%
  mutate( emoji_name = str_remove(emoji_name, '.*')) %>%
  count(emoji, emoji_name) %>%
  # Plot top 15 Emoji
  top_n(10, n) %>%
  # Create an image URL with Emoji UNICODE
  arrange(desc(n)) %>%
  mutate( emoji_url = map_chr(emoji,
                              ~paste0( 'https://abs.twimg.com/emoji/v2/72x72/'
, as.hexmode(utf8ToInt(.x)), '.png'))
  )

# ----- Plot of the Ranking of the most used Emojis -----
--

plotEmojis %>%
  ggplot(aes(x=reorder(emoji_name, n), y=n)) +
  geom_col(aes(fill=n), show.legend = FALSE, width = .2) +
  geom_point(aes(color=n), show.legend = FALSE, size = 3) +
  geom_image(aes(image=emoji_url), size=.045) +
  scale_fill_gradient(low='#2b83ba',high='#d7191c') +
  scale_color_gradient(low='#2b83ba',high='#d7191c') +
  ylab('Emoji count') +
  xlab('Names of emojis') +
  ggtitle('Most used Emojis') +
  coord_flip() +
  theme_minimal() +
  theme()

library("kableExtra")

# Extract Emojis
emoji_chat <- chat %>%
  unnest(c(emoji, emoji_name)) %>%
  mutate( emoji = str_sub(emoji, end = 1)) %>%
  mutate( emoji_name = str_remove(emoji_name, ".*"))

# Tokenize Emoji Names
```

```
emoji_chat <- emoji_chat %>%
  select(author, emoji_name) %>%
  unnest_tokens(input=emoji_name, output=emoji_words)

# Get another Lexicon with name of feelings
lexico_sentiment <- get_sentiments("nrc")
emoji_emotion <- chat %>%
  select( emoji, emoji_name) %>%
  unnest( c(emoji, emoji_name)) %>%
  mutate( emoji = str_sub(emoji, end = 1)) %>%
  mutate( emoji_name = str_remove(emoji_name, ".*")) %>%
  unnest_tokens(input=emoji_name, output=emoji_words) %>%

# Remove classification pf NEGATIVE/POSITIVE
inner_join(lexico_sentiment, by=c("emoji_words"="word")) %>%
filter(!sentiment %in% c("negative","positive")) %>%

# Keep only the 4 most frequent Emoji for each Feeling
count(emoji, emoji_words, sentiment) %>%
group_by(sentiment) %>%
top_n(4,n) %>%
slice(1:4) %>%
ungroup() %>%
select(-n)

# Putting tables Together
bind_cols(
  slice(emoji_emotion, 01:16),
  slice(emoji_emotion, 17:32)) %>%
kable() %>%
kable_styling(full_width = F, font_size = 11)

# Join with Emoji
sentiment_chat <- emoji_chat %>%
  inner_join(lexico_sentiment, by=c("emoji_words"="word")) %>%
  # Remove POSITIVE / NEGATIVE classification
  filter(!sentiment %in% c("negative","positive"))

# ----- Plot of most expressed Emotion -----
--
```

```
expressed_emo <- sentiment_chat %>%
  count(sentiment) %>%
  ggplot(aes(x=reorder(sentiment,n), y=n)) +
  geom_col(aes(fill=n), show.legend = FALSE, width = .1) +
  geom_point(aes(color=n), show.legend = FALSE, size = 3) +
  coord_flip() +
  ylab("Number of Times Expressed") + xlab("Emotion") +
  scale_fill_gradient(low="#2b83ba",high="#d7191c") +
  scale_color_gradient(low="#2b83ba",high="#d7191c") +
  ggtitle("Most frequently expressed emotion", "Expressed by use of emojis") +
  theme_minimal()

ggplotly(expressed_emo)

#----- Most Frequently expressed Emotion by Users -----
--

frequent_emo <- sentiment_chat %>%
  count(author, sentiment) %>%
  left_join(filter(lexico_sentiment, sentiment %in% c("negative","positive")),
by=c("sentiment"="word")) %>%
  rename( feeling = sentiment.y) %>%
  mutate( feeling = ifelse(is.na(feeling), "neutral", feeling)) %>%
  mutate( feeling = factor(feeling, levels = c("negative", "neutral", "positive"), ordered=T) ) %>%
  group_by(author) %>%
  top_n(n = 8, n) %>%
  slice(1:8) %>%
  ggplot(aes(x = reorder(sentiment, n), y = n, fill = feeling)) +
  geom_col() +
  scale_fill_manual(values = c("#d7191c", "#fdae61", "#1a9641")) +
  ylab("Number of Times Expressed") +
  xlab("Emotion") +
  coord_flip() +
  facet_wrap(~author, ncol = 3, scales = "free_x") +
  ggtitle("Most frequently expressed emotion ", "Expressed by use of emojis") +

  theme_minimal() + theme(legend.position = "bottom")

ggplotly(frequent_emo)
```

```
# -----  
# Targeting Advertisement using WhatsApp Chat  
# -----  
  
chat$text =tolower(chat$text)  
  
# Unnesting Text  
messages <- chat %>%  
  unnest(text) %>%  
  count(author, text, sort = TRUE) %>%  
  group_by(author)  
  
messages <- subset(messages, select=-c(n))  
  
# Searching for Companies Name used by Users  
keyword <- list("google", "amazon", "microsoft", "linkedin", "youtube", "ipl",  
               "delight", "films", "dream11", "geeksforgeeks", "coursera",  
               "udemy")  
  
# Finding Authors who have used the following Keywords  
find_keyword <-  
  messages$author[grepl(keyword[[1]], messages$text)]  
  
# Finding the Text Messages in which the following Keywords are used  
find_message <-  
  messages$text[grepl(keyword[[1]], messages$text)]  
  
ef = data.frame(find_keyword, find_message, company="google")  
  
for (i in keyword) {  
  find_keyword <-  
    messages$author[grepl(i, messages$text)]  
  find_message <-  
    messages$text[grepl(i, messages$text)]  
  ff <- data.frame(find_keyword, find_message, company=i)  
  ef <- rbind(ef, ff)  
}  
  
# Searching for Products Name used by Users
```



```
product <- list("classroom", "pizza", "teams", "collab", "livestream", "cricket",
               "avengers", "python", "scholarship", "javascript", "sql", "aws",
               "connections", "games")

# Finding Authors who have used the following Product
find_message <-
  messages$text[grepl(product[[1]], messages$text)]

# Finding the Text Messages in which the following Product are used
author <-
  messages$author[grepl(product[[1]], messages$text)]

gf = data.frame(author, find_message, product="classroom")

for(j in product) {
  author <-
    messages$author[grepl(j, messages$text)]
  find_message <-
    messages$text[grepl(j, messages$text)]
  bf <- data.frame(author, find_message, product=j)
  gf <- rbind(gf, bf)
}

# Merging
companies_product <- merge(x=ef, y=gf, by="find_message", all.y = TRUE)
companies_product <- subset(companies_product, select=c(find_message, find_keyword))
companies_product <- data.frame(author=companies_product$author, company=companies_product$company, product=companies_product$product)

# Finding the Text Messages where both Company1 and Services are Mentioned in Same
company1<-
list("google", "delight", "microsoft", "google", "coursera", "coursera",
     "udemy", "ipl", "linkedin", "dream11", "geeksforgeeks", "ipl")
service <- list("classroom", "pizza", "teams", "collab", "javascript", "sql",
               "aws", "cricket", "connections", "games", "python", "teams")
```

```
ads_target <- companies_product %>%
  filter(company=="google" & product=="classroom")

j=1
for(i in company1) {
  ads1 <- companies_product %>%
    filter(company==i & product==service[j])
  ads_target <- rbind(ads_target, ads1)

  j=j+1
}

ads_target <- ads_target %>%
  count(author, company, product, sort=TRUE)

# Targeting Ads on the basis of Product Name used by User
comapany_product <- distinct(companies_product)

# Targeting Ads on the basis of Company and Product Name both used by User
ads_target <- distinct(ads_target)

# ----- Plotting Data -----
--

ggplotly (
  ads_target %>%
    group_by(author) %>%
    top_n(n=5, n) %>%
    arrange(author, desc(n)) %>%
    mutate(order=row_number()) %>%
    ggplot(aes(x=reorder(product, n), y=n, fill=author, color=author)) +
    geom_col(show.legend=FALSE, width=.1) +
    geom_point(show.legend=FALSE, size=3) +
    ylab('Number of time mentioned by Author') +
    xlab('Company Service') +
    coord_flip() +
    facet_wrap(~author, ncol = 3, scales = 'free') +
    ggtitle('Most favourable Advertisment suggestion') +
    theme_minimal()
)
```

```
# -----  
# ADVERTISEMENT CLICK PREDICTION  
# -----  
  
# Goal of the part is to predict if a particular user is likely to click on pa  
rticular ad or not based on his feature.  
ads <- read.csv("C:/Users/Kiran Maharana/Documents/SEM IV/Summer Internship/ad  
vertising.csv",header=T)  
  
ad = subset(ads, select = -c(Ad.Topic.Line, City, Country, Timestamp) )  
  
# Checking for Duplicates  
duplicated(ads)  
  
# Attribute Type Classification  
# Determining the type of attributes in the given Dataset  
numeric_columns = c('Daily Time Spent on Site', 'Age', 'Area Income', 'Daily I  
nternet Usage' )  
categorical_columns = c( 'Ad Topic Line', 'City', 'Male', 'Country', 'Clicked  
on Ad' )  
  
# Exploratory data analysis  
# ----- What age group does the Dataset majorly consist of? -----  
png(file = "histogram.png")  
  
# Create the histogram.  
hist(ads$Age, xlab = "Age", col = "yellow", border = "blue")  
  
# Save the file.  
dev.off()  
  
# Here, we can see that most of the internet users are having age in the range  
of 26 to 42 years  
  
sprintf('Age of the oldest person is: %d years', max(ads$Age))  
sprintf('Age of the youngest person is: %d years', min(ads$Age))  
sprintf('Average age in dataset is: %f years', mean(ads$Age))
```

```
# ----- What is the income distribution in different age groups? -----  
--  
ggplot(ads, aes(x = Age, y = Area.Income)) +  
  geom_point(color="green")+  
  ggtitle("Age vs Income") +  
  theme_minimal()  
  
# Here, we can see that mostly teenagers are higher earners with age group of  
# 20-40 earning 50k-70k  
  
# ----- Which age group is spending maximum time on the internet? -----  
--  
ggplot(ads, aes(x = Age, y = Daily.Internet.Usage)) +  
  geom_point(color="blue")+  
  ggtitle("Daily Internet Usage") +  
  theme_minimal()  
  
# From the above plot its evident that the age group of 25-  
# 40 is most active on the internet  
  
# ----- Which gender has clicked more on online ads? -----  
--  
ads1 <- ads %>% group_by(Male) %>% filter(Clicked.on.Ad==1) %>% summarise(clicks_count=n()) %>% arrange(desc(clicks_count))  
  
# Based on above data we can see that a greater number of females have clicked  
# on ads compared to male.  
  
# Maximum number of internet users belong to which country in the given dataset?  
ads2 <- ads %>% group_by(Country) %>% summarise(number_of_users=n()) %>% arrange(desc(number_of_users))  
  
# Based on the above data frame we can observe that maximum number of users are from France and Czech  
  
# ----- Did we match our baseline that we set? -----  
--  
ads3 <- ads %>% group_by(Clicked.on.Ad) %>% summarise(Clicked_on_Ad=mean(Clicked.on.Ad),Daily_Time_Spent_on_Site=mean(Daily.Time.Spent.on.Site),Age=mean(Age
```

```
),Area_Income=mean(Area.Income),Daily_Internet_Usage=mean(Daily.Internet.Usage
))

# ----- What is the relationship between different features? -----
--
library(psych)
pairs.panels(ads)

# Data Cleaning
library(naivebayes)
library("nanian")
vis_miss(ads,cluster=TRUE)

# Data Model Implementation
str(ad)

ad$Daily.Time.Spent.on.Site <- as.factor(ad$Daily.Time.Spent.on.Site)
ad$Age <- as.factor(ad$Age)
ad$Area.Income <- as.factor(ad$Area.Income)
ad$Daily.Internet.Usage <- as.factor(ad$Daily.Internet.Usage)
ad$Male <- as.factor(ad$Male)
ad$Clicked_Ad <- as.factor(ad$Clicked.on.Ad)
ad = subset(ad, select = -c(Clicked.on.Ad) )

set.seed(1234)
ind <- sample(2, nrow(ad), replace = T, prob = c(0.8, 0.2))
train <- ad[ind == 1,]
test <- ad[ind == 2,]

# ----- Naive Bayes Model -----
--
# -----
--

model <- naive_bayes(Clicked_Ad ~ ., data = train)
model

plot(model)

# Predict of Train Data
```

```
p <- predict(model, train, type = 'prob')
v<- cbind(p, train)
```

```
# Predict of Test Data
```

```
m<- predict(model, test, type = 'prob')
n<- cbind(m,test)
```

```
# Confusion Matrix - train data
```

```
p1 <- predict(model, train)
(tab1 <- table(p1, train$Clicked_Ad))
```

```
# Misclassification
```

```
print("Misclassification of Train Data: ")
1 - sum(diag(tab1)) / sum(tab1)
```

```
# Confusion Matrix - test data
```

```
p2 <- predict(model, test)
(tab2 <- table(p2, test$Clicked_Ad))
```

```
# Misclassification
```

```
print("Misclassification of Test Data: ")
1 - sum(diag(tab2)) / sum(tab2)
```

```
# Model Evaluation
```

```
library(caret)
```

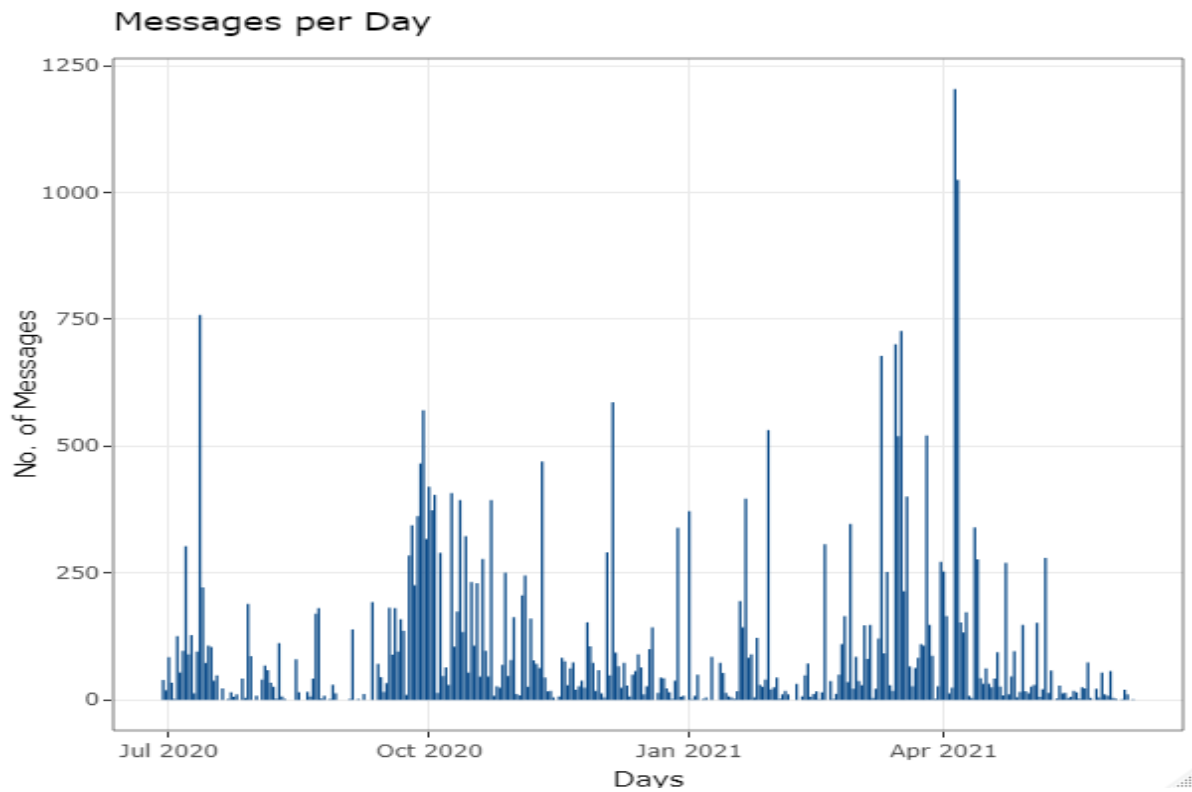
```
cm <- table(test$Clicked_Ad, p2)
cm
```

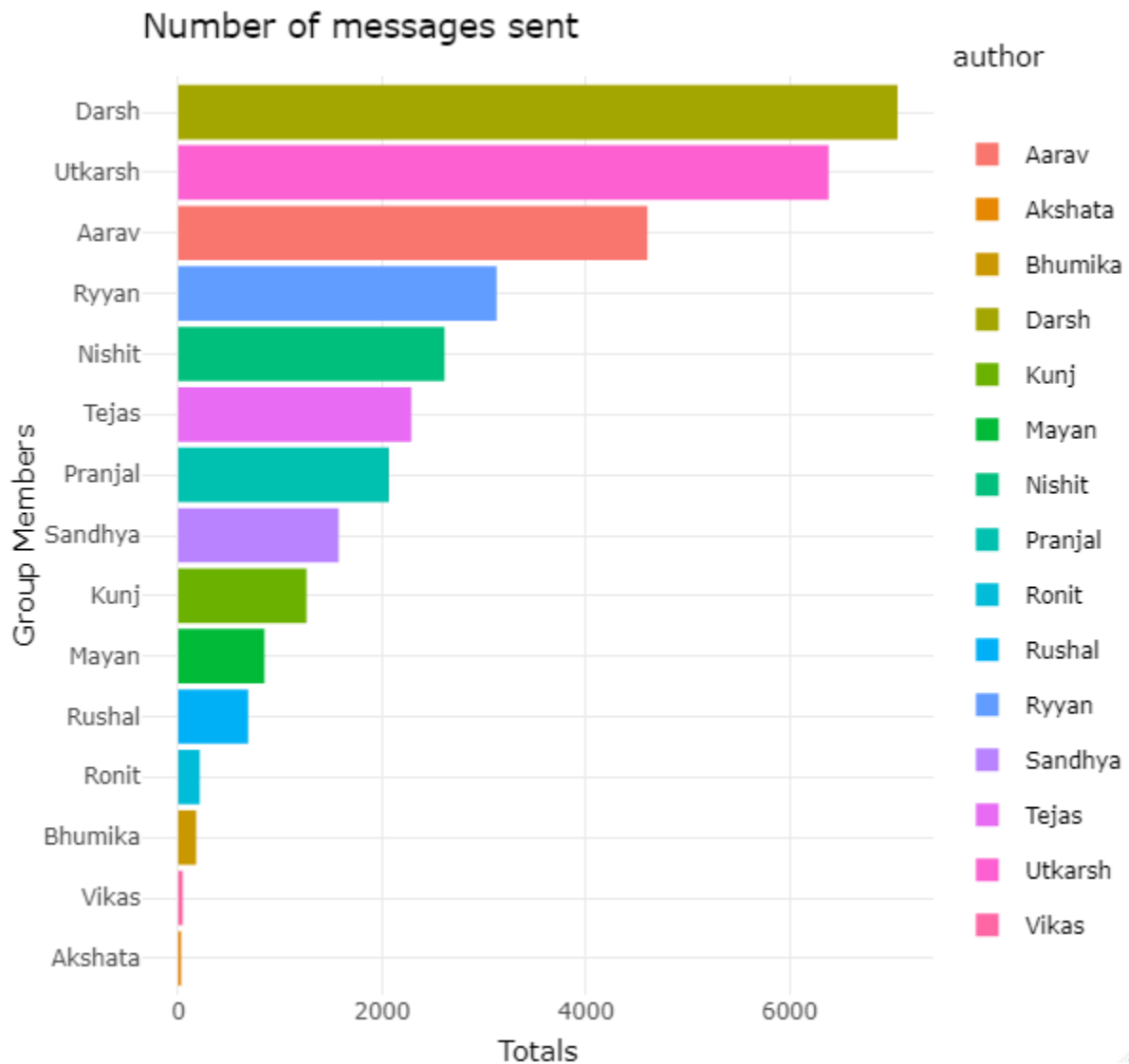
```
confusionMatrix(cm)
```

```
# ----- Decision Tree -----
--
# -----
--
```

```
library(party)
```


Result:

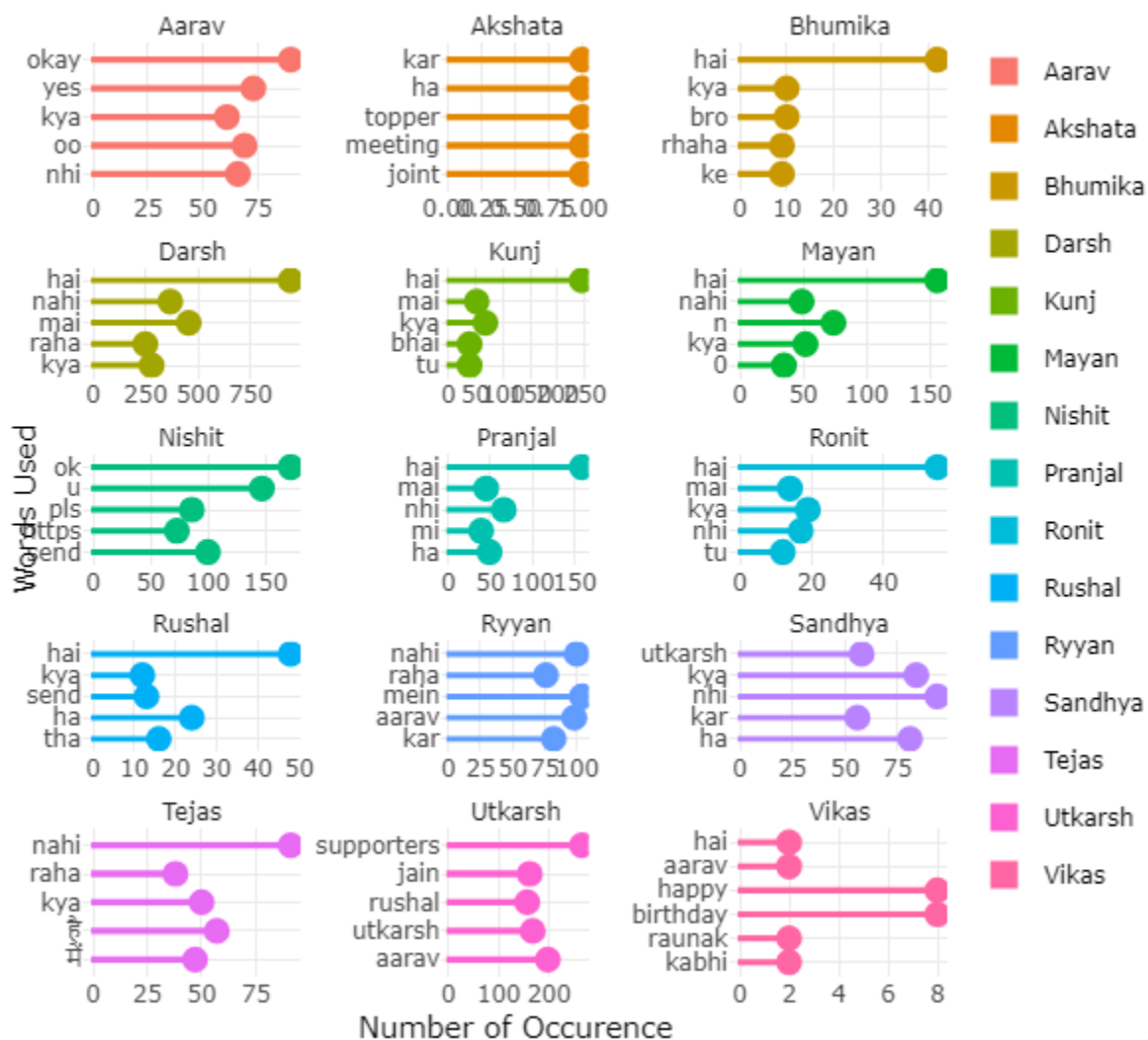


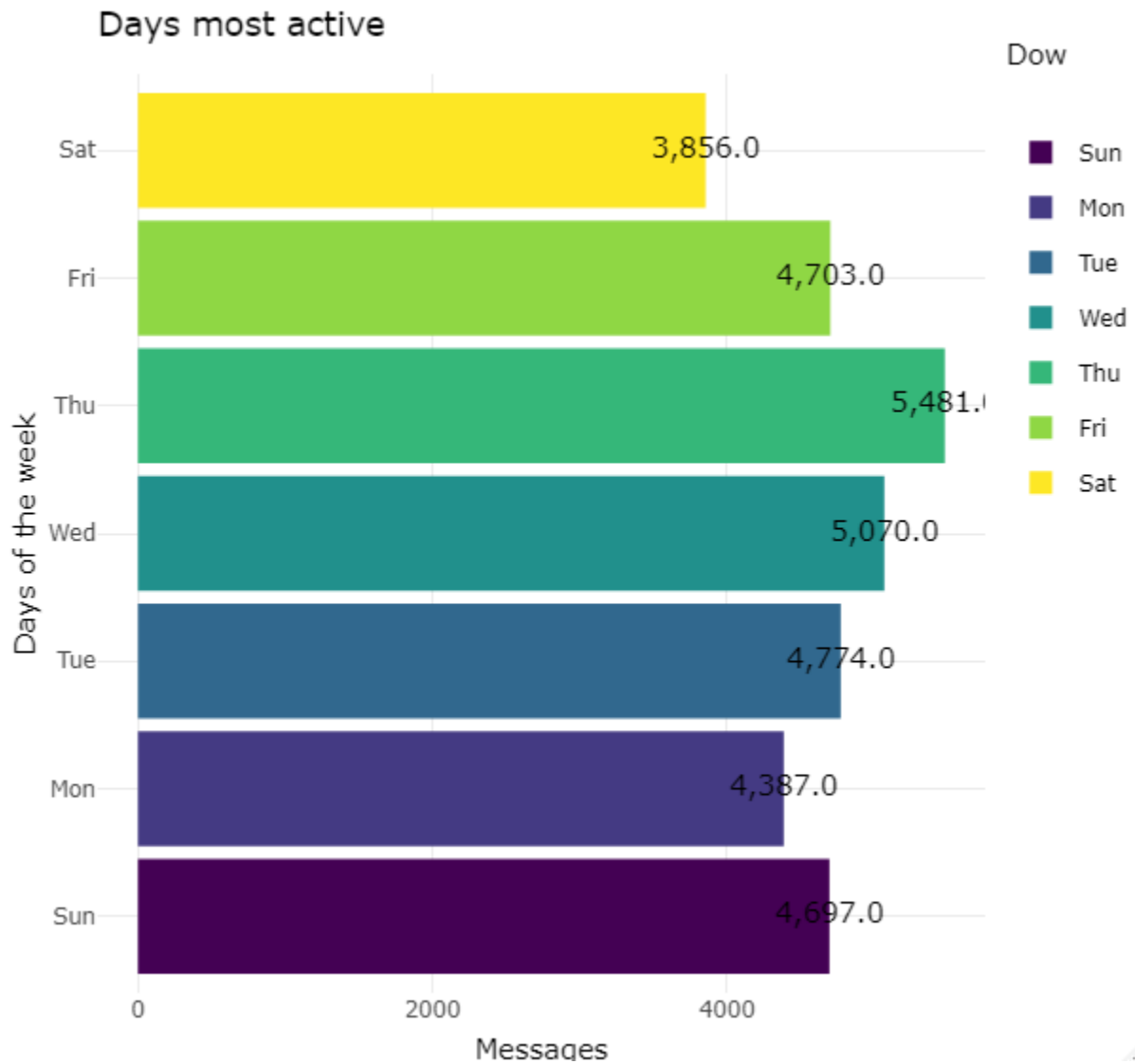


Most used emojis by users

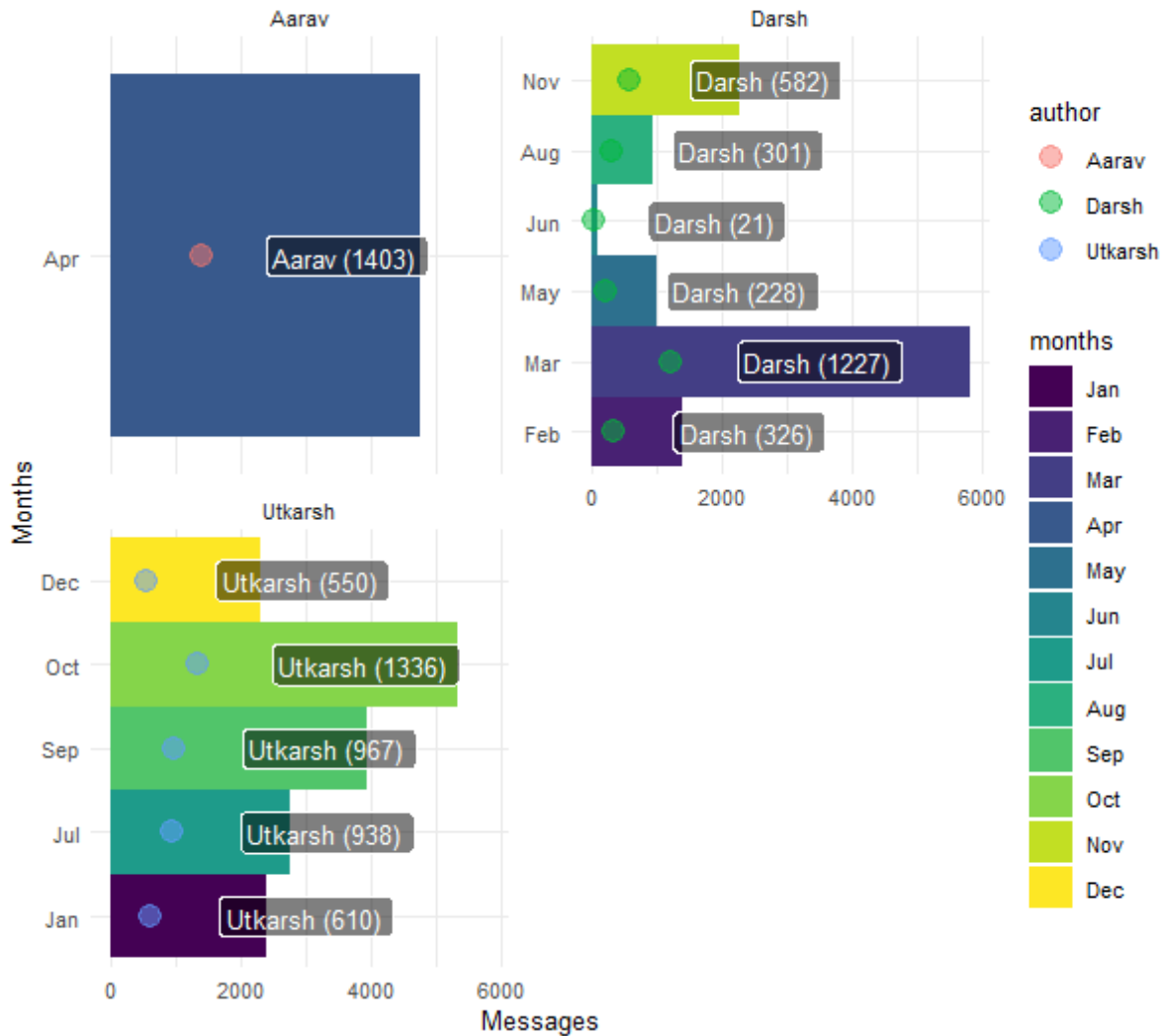


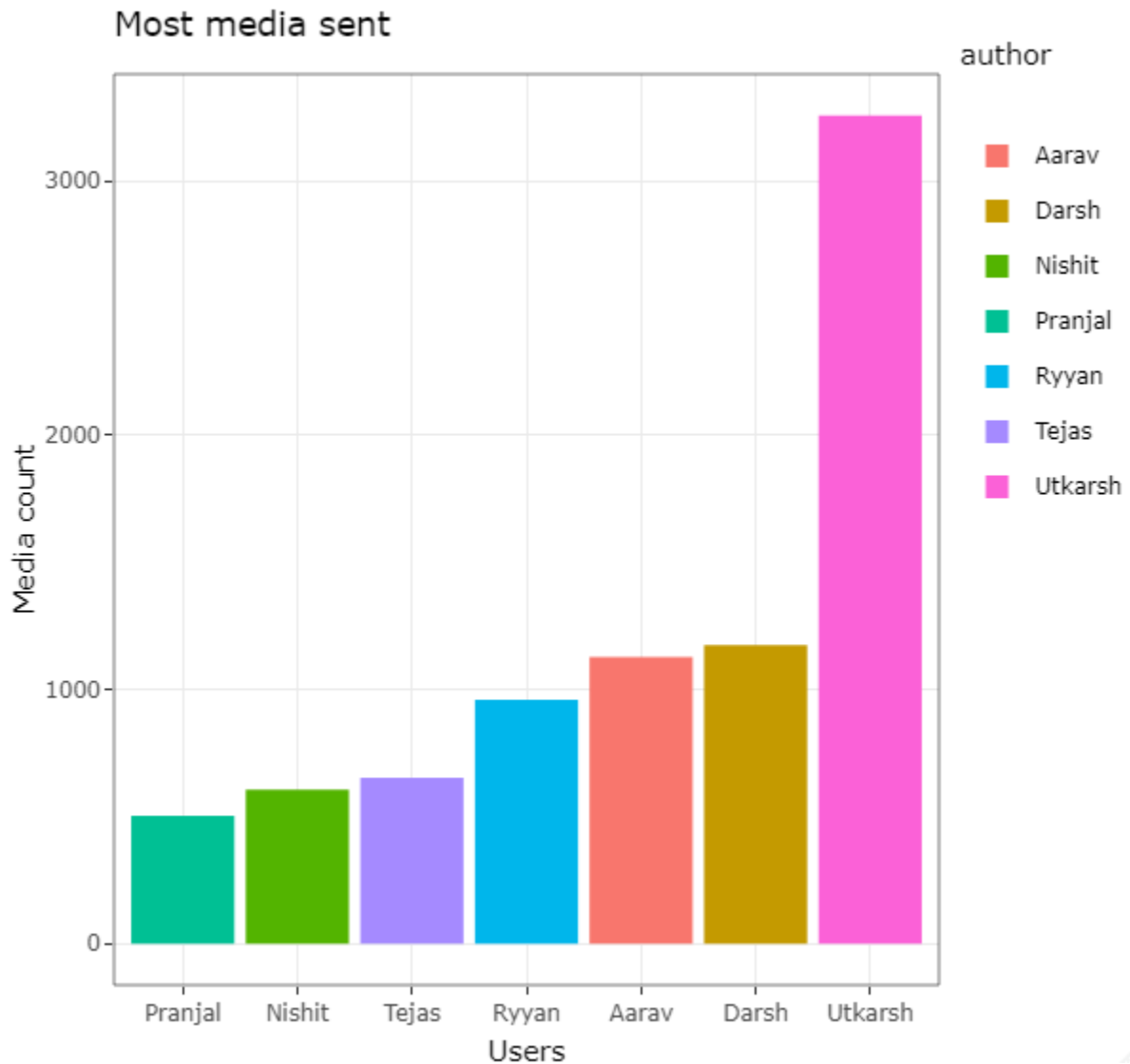
Most Words used by Users

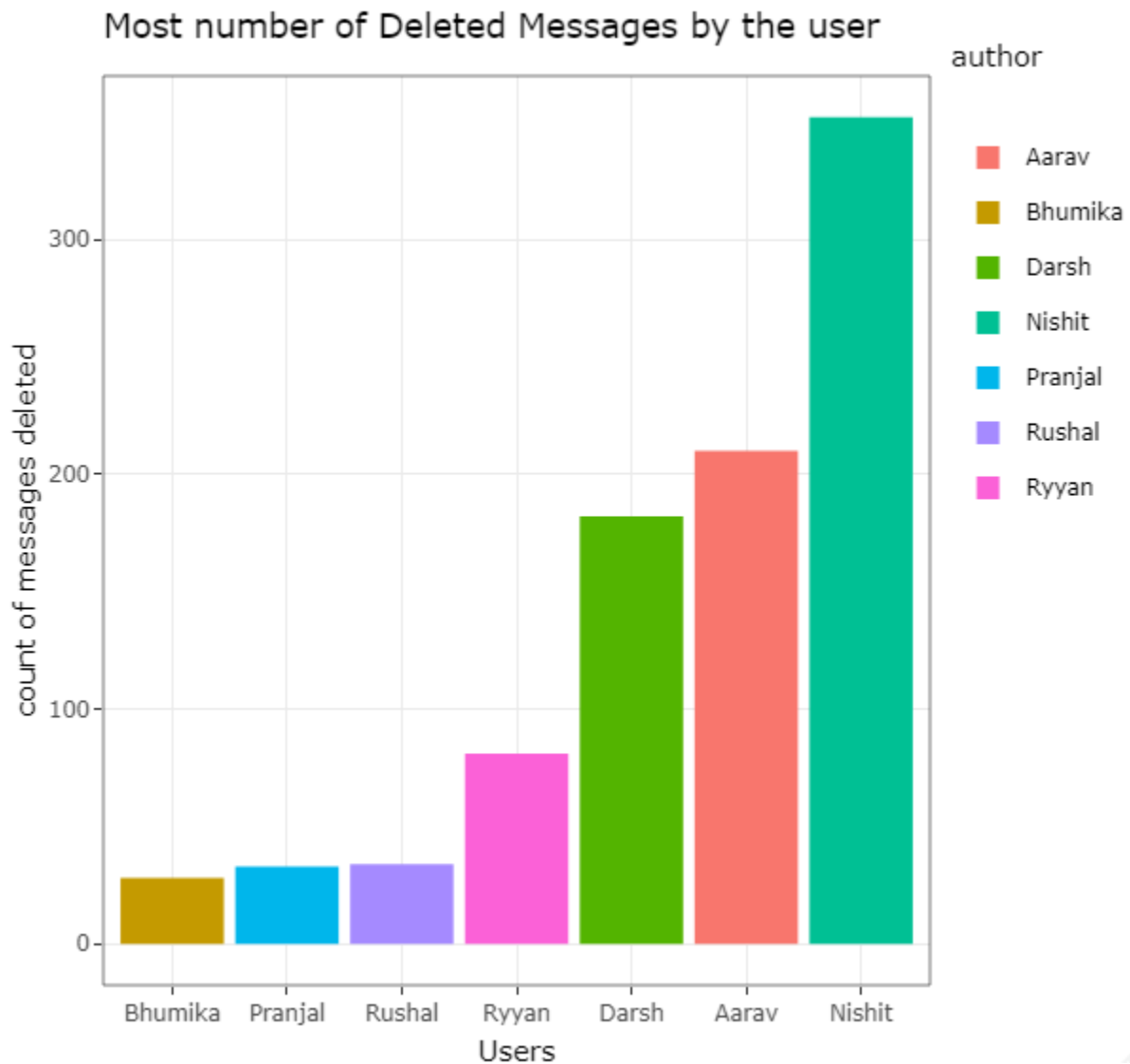




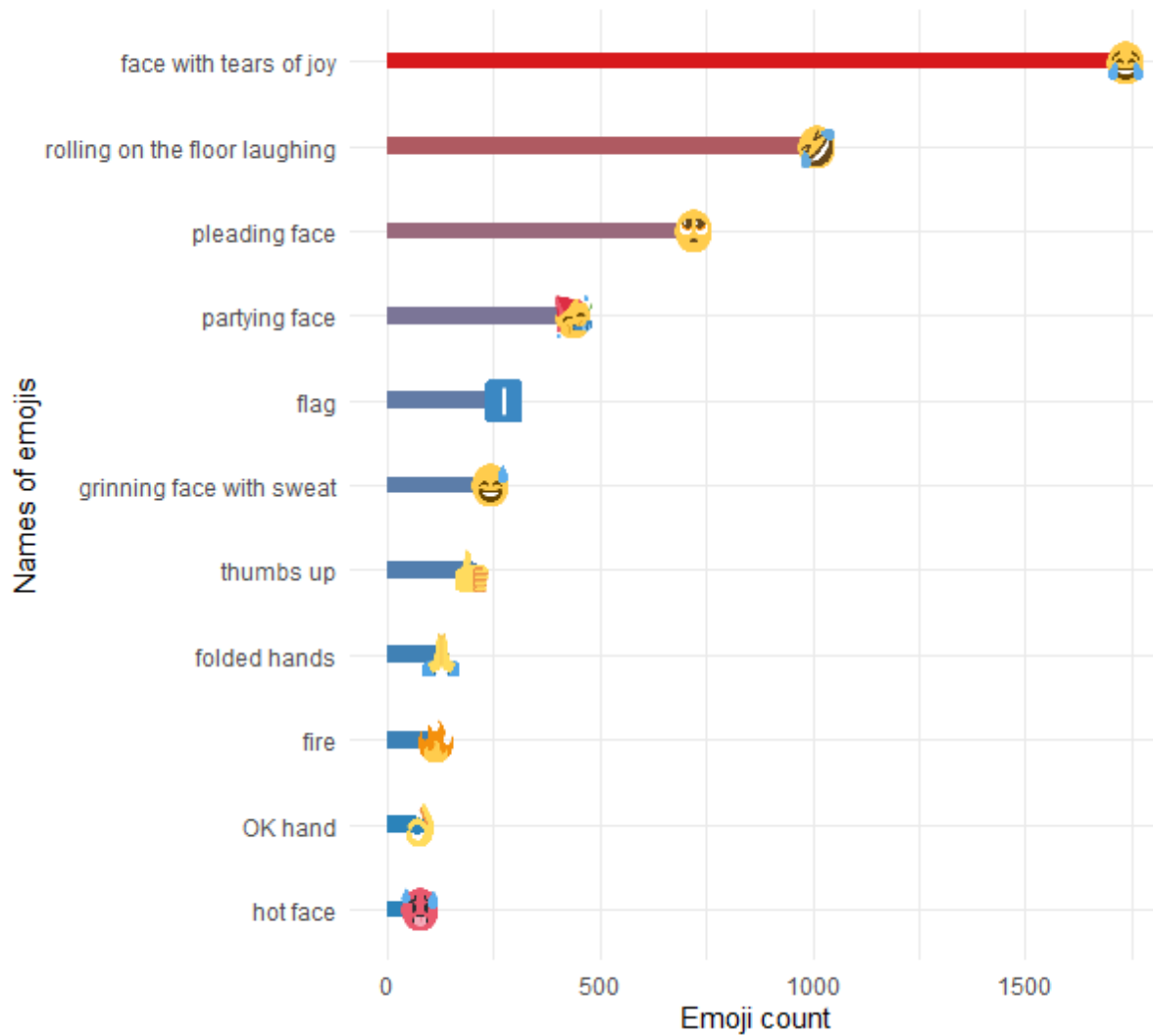
Most Active Member Each Month

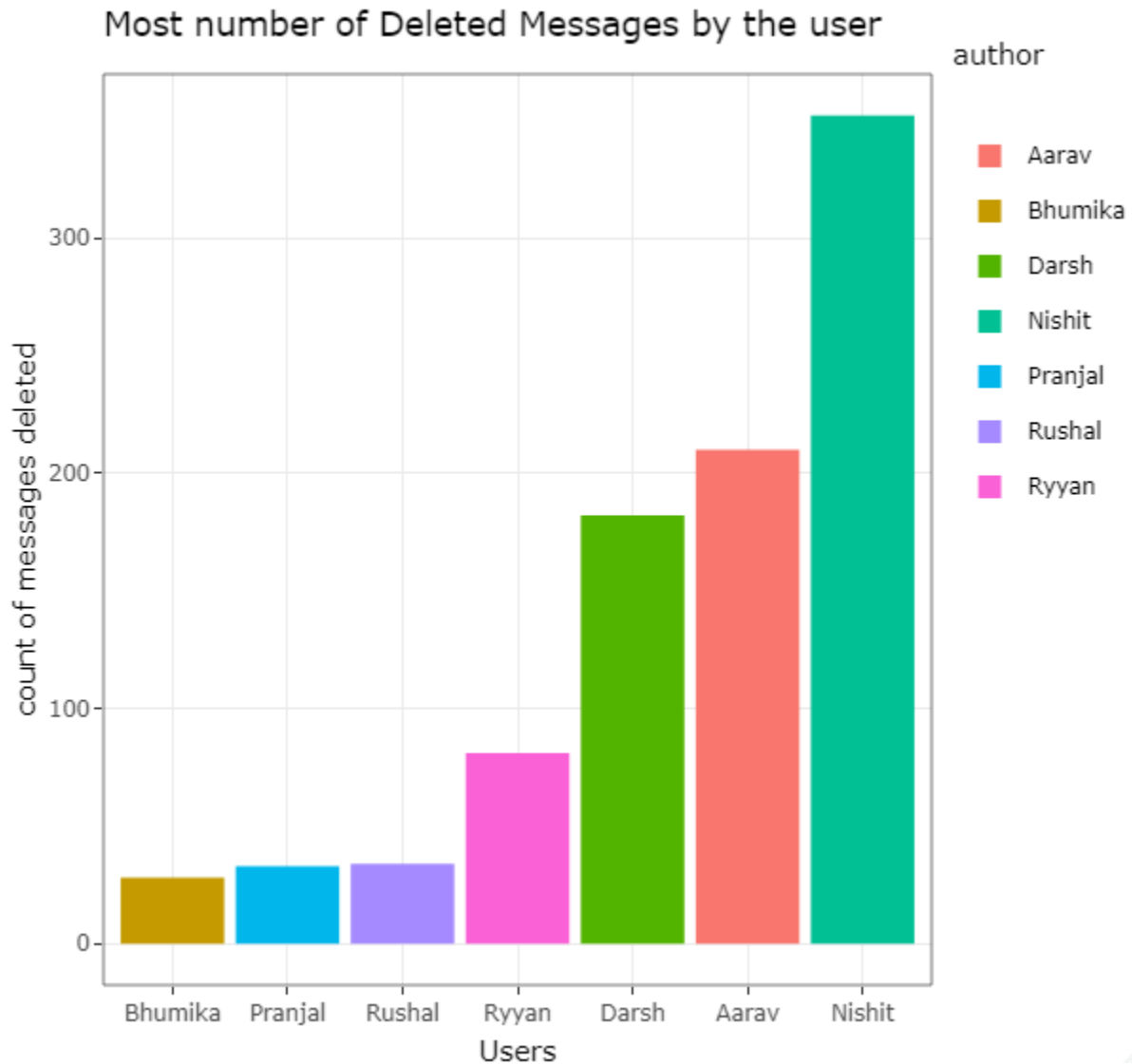






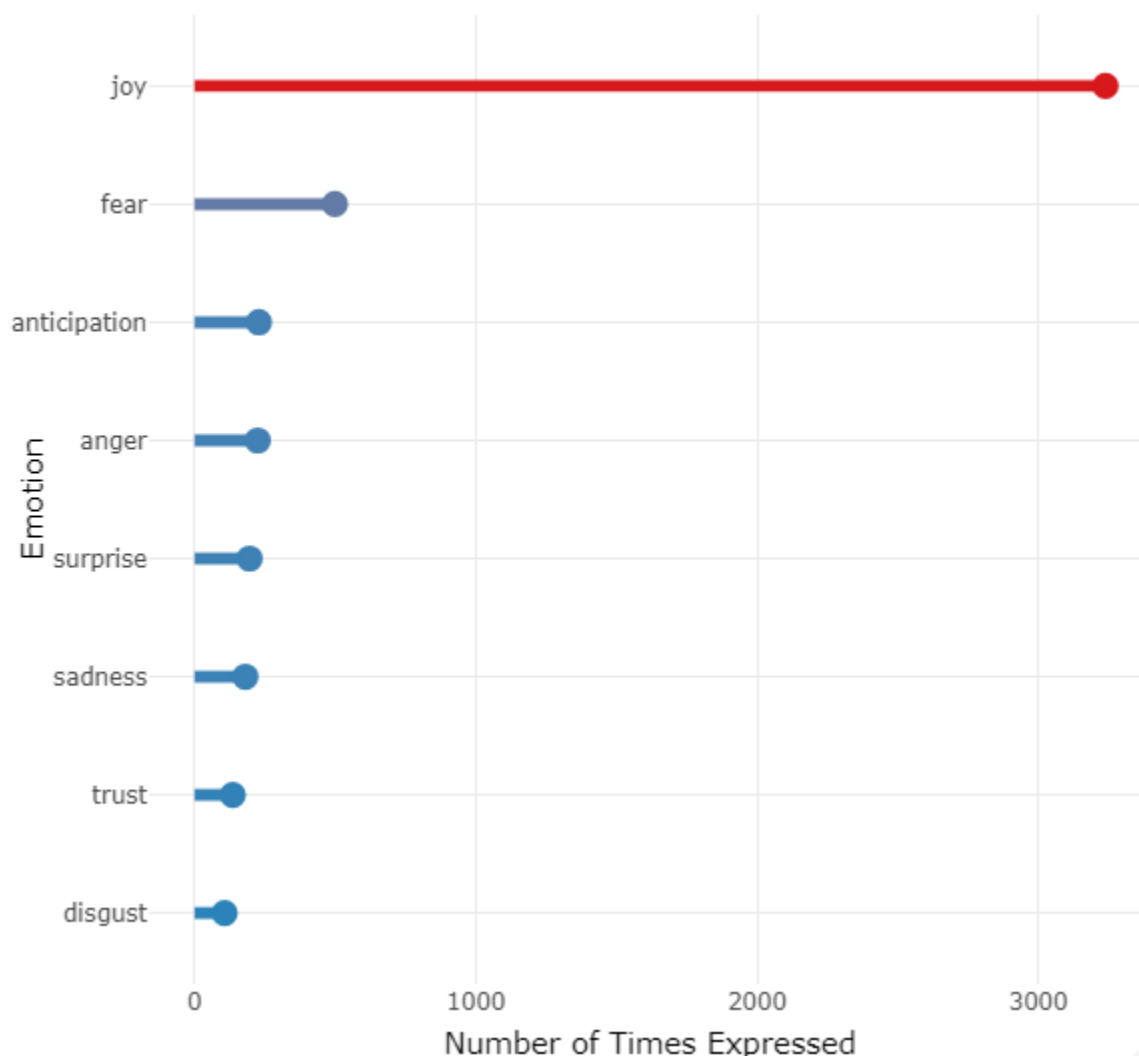
Most used Emojis

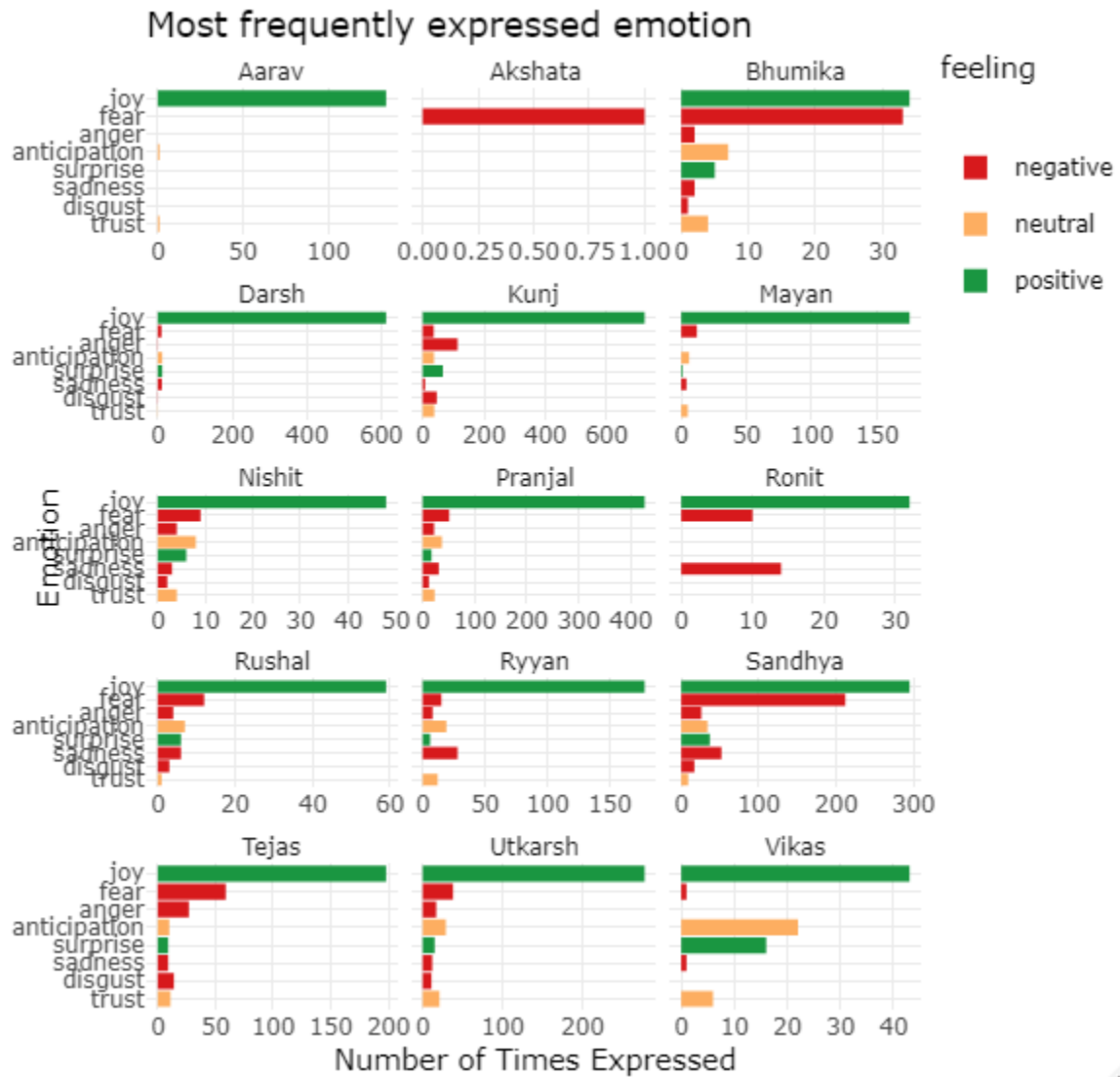




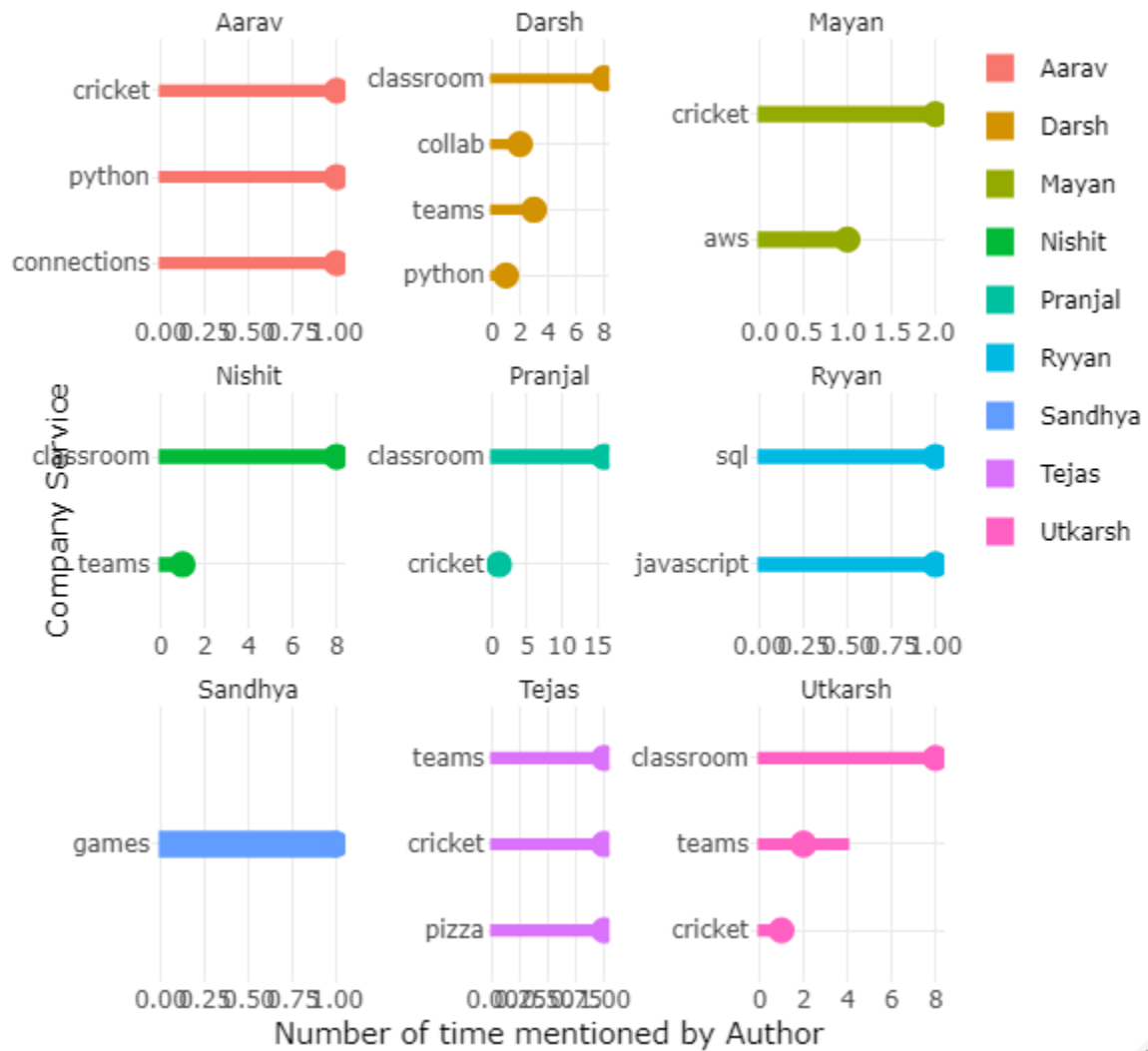
emoji...1	emoji_words...2	sentiment...3	emoji...4	emoji_words...5	sentiment...6
<U+0001F4A5>	collision	anger	<U+0001F382>	birthday	joy
<U+0001F631>	fear	anger	<U+0001F602>	joy	joy
<U+0001F631>	screaming	anger	<U+0001F923>	laughing	joy
<U+0001F911>	money	anger	<U+0001F929>	star	joy
<U+0001F382>	birthday	anticipation	<U+0001F614>	pensive	sadness
<U+0001F601>	beaming	anticipation	<U+0001F622>	crying	sadness
<U+0001F911>	money	anticipation	<U+0001F62D>	crying	sadness
<U+0001F929>	star	anticipation	<U+25AA>	black	sadness
<U+0001F40D>	snake	disgust	<U+0001F382>	birthday	surprise
<U+0001F61E>	disappointed	disgust	<U+0001F62E>	mouth	surprise
<U+0001F624>	nose	disgust	<U+0001F911>	money	surprise
<U+0001F631>	screaming	disgust	<U+0001F911>	mouth	surprise
<U+0001F40D>	snake	fear	<U+0001F60B>	food	trust
<U+0001F525>	fire	fear	<U+0001F911>	money	trust
<U+0001F605>	sweat	fear	<U+0001F929>	star	trust
<U+0001F613>	sweat	fear	<U+270C>	victory	trust

Most frequently expressed emotion

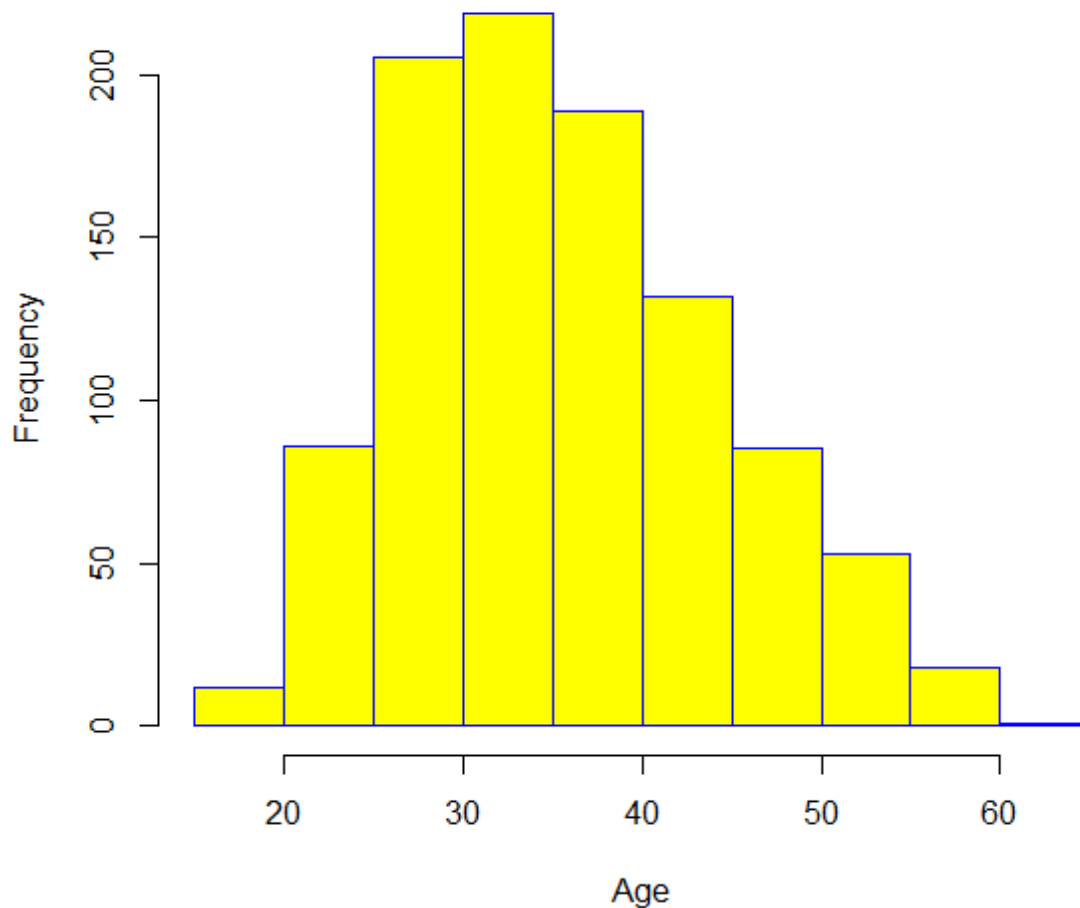




Most favourable Advertisement suggestion

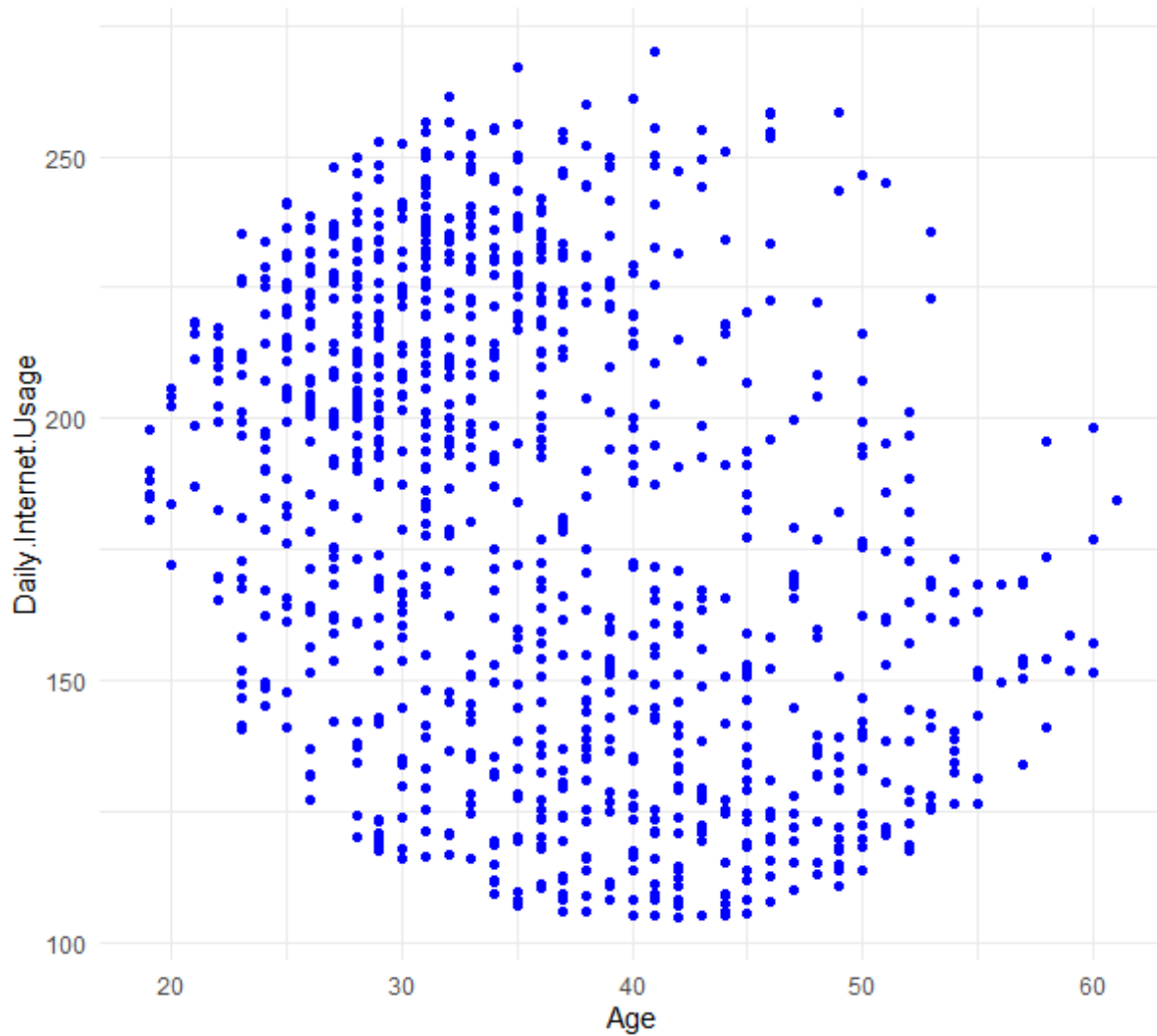


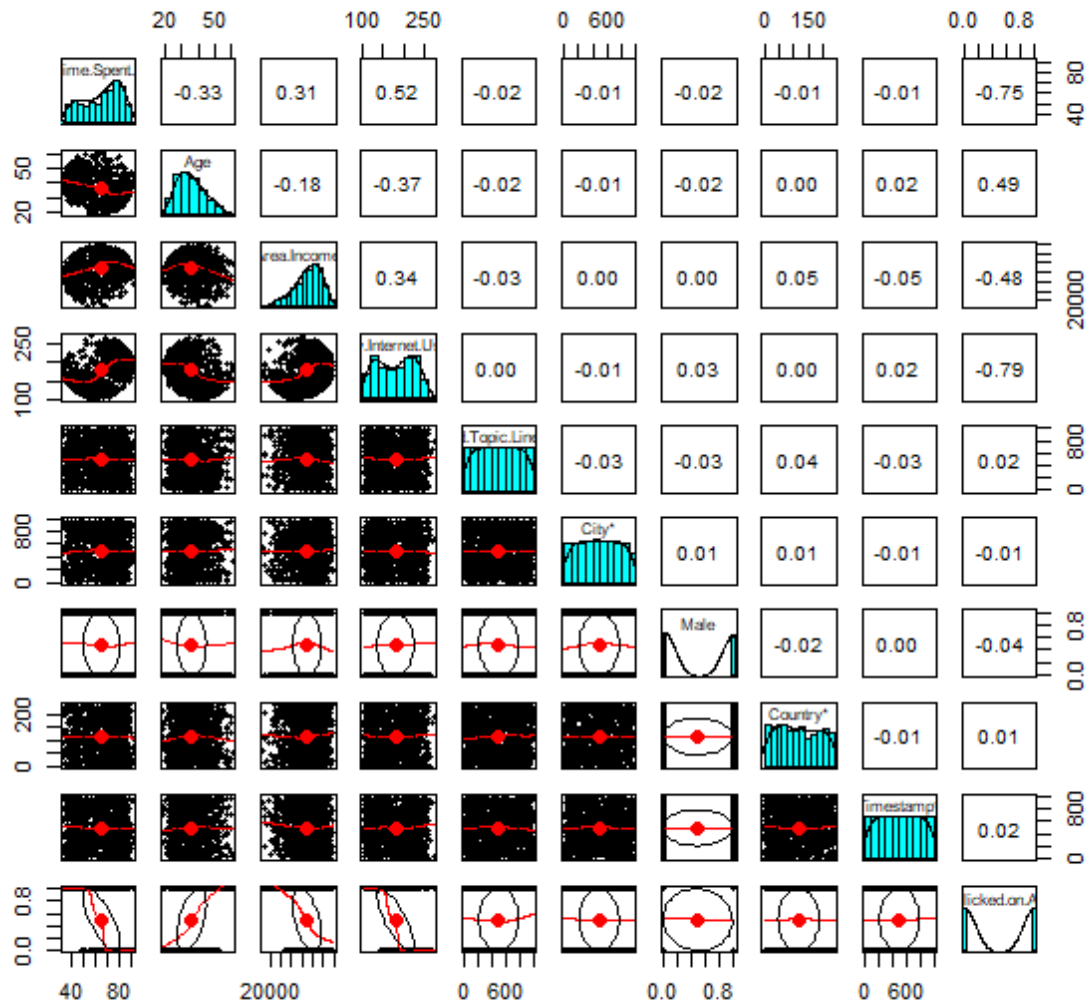
Histogram of ads\$Age

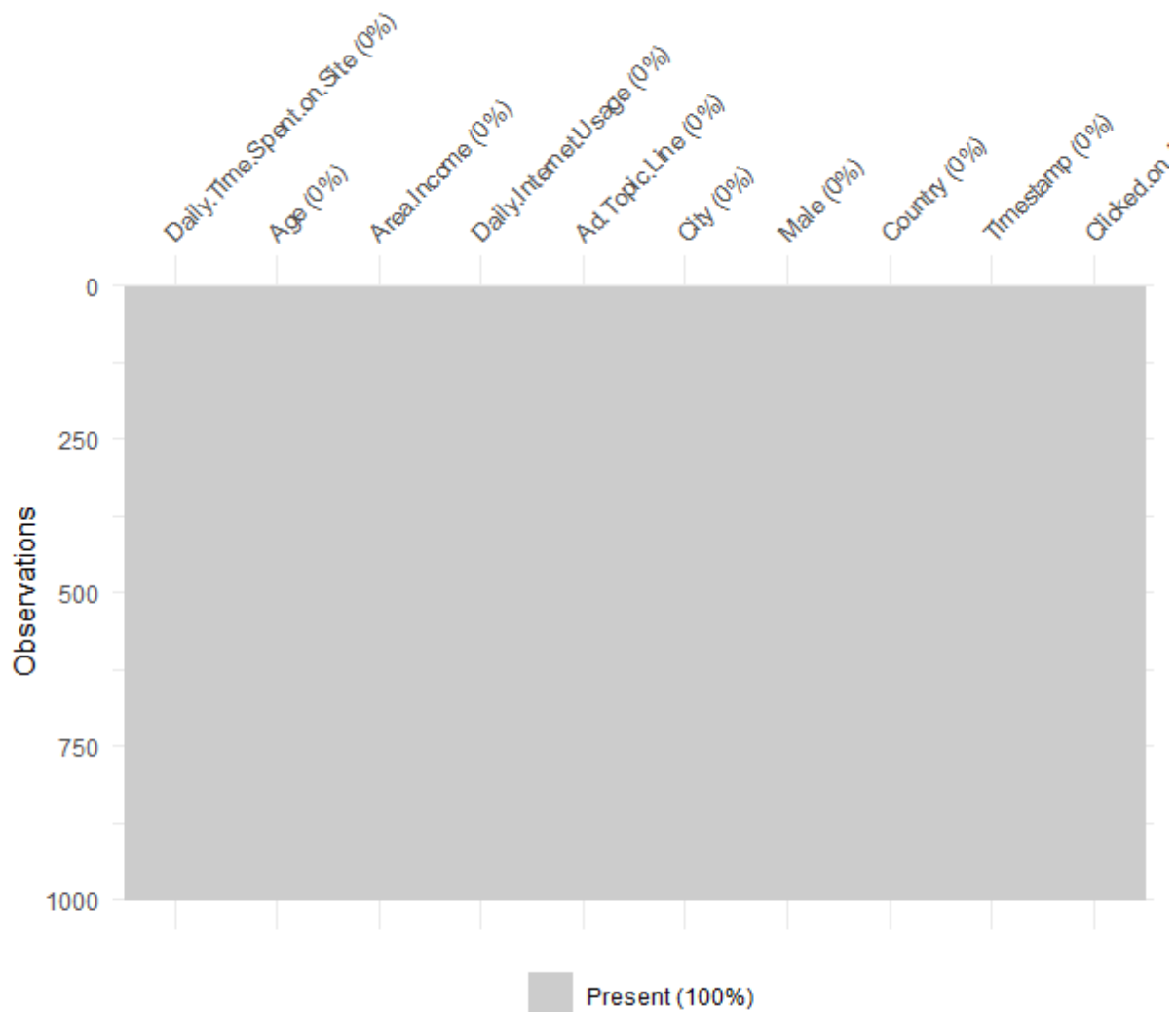




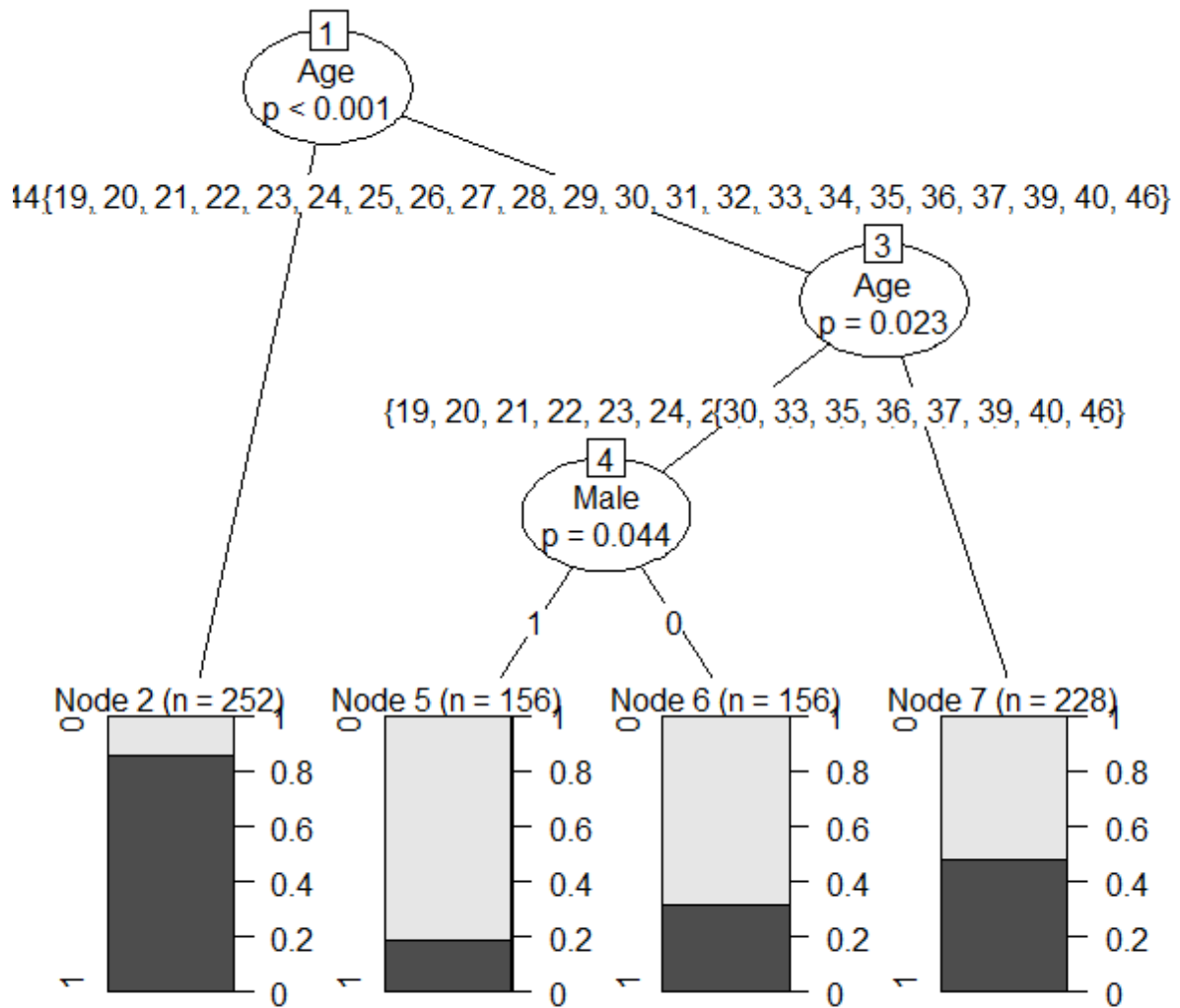
Daily Internet Usage







		0	1
Male	0		
	1		



Conclusion:

We analyse WhatsApp chat where we have depicted interesting bar graphs. Here, we have analysed activity of a person within the group chat and we learned how active he/she is on group, how often he share media files or shares stickers, how many messages have he/she sent and many more. Exploratory Data Analysis, is to apply a sentiment analysis algorithm which provides positives, negative and neutral part of the chat. We learned about the most frequently expressed emotion by Users on a WhatsApp Group such as Joy, Fear, Anger, Anticipation, Surprise, Sadness, Disgust and Trust.

Online advertising is a multi-billion-dollar industry that has served as one of the great success stories for machine learning. We have extracted keyword related to Companies their Services or Products with frequency of mention. This provides us a great insight about how interested and sentiments of User related to product or service. We have learned about Click-Ad prediction and understood what are the factors affecting and influencing the User to click on the Advertisement.

Comparing all the above implementation models, we conclude that Naive Bayes Algorithm gives us the maximum accuracy for determining the click probability. We believe in future there will be fewer ads, but they will be more relevant. And also these ads will cost more and will be worth it.

References:

- [1]. <https://medium.com/analytics-vidhya/chat-analysis-on-whatsapp-part-2-sentiment-analysis-and-data-visualization-with-r-f148592fa1b0>
- [2]. <https://medium.com/@ismaelbouarfa/malicious-url-detection-with-machine-learning-d57890443dec>
- [3]. <https://www.kaggle.com/zohebabai/exploratory-project-on-advertising-dataset>
