SOCIAL MEDIA SENTIMENT ANALYSIS

In this project we have a data about different social media platforms like Instagram, Twitter etc

So, in our dataset we have data like No of likes, No of comments, No of Shares, User follow count, Post type (image, video, text) etc.

Here is the look of our dataset

1	Post ID	Post Content	Sentiment Label	Number of Likes	Number of Shares	Number of	User Follo	Post Date and Time	Post Type	Language
2	aa391375-	Word who nor center everything better political. Various court realize arrive.	Neutral	157	243	64	4921	10-01-2024 00:14	video	fr
3	1c9ec98d-	Begin administration population good president particularly. Some study them	Positive	166	49	121	612	03-02-2024 00:20	image	es
4	170e5b5b-	Thousand total sign. Agree product relationship several stop conference.	Positive	185	224	179	9441	25-07-2024 14:20	video	de
5	aec53496-	Individual from news third. Oil forget them different account skin.	Neutral	851	369	39	6251	20-02-2024 09:15	text	de
6	4eacddb7-	Time adult letter see reduce. Attention suddenly it. Agency eye decade art friend	Negative	709	356	52	1285	01-03-2024 04:17	image	de
7	bdc905fc-	Compare ok attack more in camera. Car better example rock within.	Positive	789	231	177	916	08-05-2024 03:20	image	zh
8	6497fe71-	Sing cost sound. Heavy on south.	Positive	639	163	175	5478	17-03-2024 07:07	video	fr
9	a8603008-	Arrive large small difference officer read. Control herself art purpose become.	Positive	828	34	161	5700	24-07-2024 13:03	image	fr
10	b6af485e-	Sit small thank thank protect. Deal employee kid very. Top scientist simply nation	Neutral	924	153	2	7115	08-01-2024 05:42	image	zh
11	eb9d1826-	Month leader subject beat.	Positive	169	22	77	6824	06-02-2024 03:30	video	en
12	3b89b4a2-	Audience away easy light federal institution. Available sign social affect now	Neutral	738	414	107	1974	28-02-2024 06:40	text	zh
13	ec16c7a5-	Many computer father yourself policy attorney. Money inside full investment	Neutral	448	363	169	3714	25-01-2024 20:16	image	en
14	732818c7-	Listen middle general over right local cup. Big mean southern music recent.	Neutral	44	58	25	9868	18-04-2024 16:27	text	es
15	a9145924-	Suggest control report south hair trial agreement. Young must stay structure	Positive	420	11	169	187	24-02-2024 14:59	video	es
16	4f757e17-	Page partner doctor white huge. Technology speech foreign activity. Catch box	Neutral	373	434	133	6266	27-02-2024 19:03	video	fr
17	1100a964-	Often blood floor might development. Score sit decade many on story. Color	Negative	692	175	69	886	24-03-2024 23:24	text	zh
18	63f4c673-	What its employee prevent poor surface economic. Surface special sit	Positive	736	331	159	4993	11-03-2024 01:20	image	zh
19	b1b6a8e8-	Go fine strong newspaper expert you. Civil sing character.	Positive	243	55	155	8591	29-05-2024 07:03	image	zh
20	78c5424e-	Rule dream begin. Lot hotel environment here. Energy out pass environment.	Positive	992	346	179	674	05-06-2024 02:35	video	zh
21	879bb079-	Continue better trial thought could. Population arm task audience thought prove	Positive	454	409	35	5231	01-07-2024 07:15	video	de
22	7cc0c316-	Final first increase throw become state per. Language force expect resource	Positive	625	84	107	8846	04-05-2024 03:15	image	de
23	8fedf6ca-	One building performance point anything. Support purpose me his democratic	Neutral	291	476	26	9255	20-01-2024 16:09	text	es

Our goal is to analyse this dataset and produce some insights by using some machine learning algorithms like linear regression, logistic regression.

There is a chance that we encounter the methods and libraries that we didn't used until now. No worries we will learn and go.

So, Let's get started…. Shall we

Be it any project first we have to import the dataset into the RStudio. To do that we have read.csv()

And if you don't want to give path we can choose the path of the dataset everytime using file.choose().

```
SocialMediaData <- read.csv(file.choose())
```

So, After successfully importing the dataset what we should do everytime. We have to check for null values.

To do that we have is.na() method to say whether the entity is null or not. And to remove the null values we have na.omit().

```
sum(is.na(SocialMediaData))

SocialMediaData <- na.omit(SocialMediaData)
```

```
sum(is.na(SocialMediaData))
1] 0
SocialMediaData <- na.omit(SocialMediaData)
```

So now all ready with the dataset we have to preview the dataset and analyse the summary of the dataset.

```
head(SocialMediaData)

str(SocialMediaData)

summary(SocialMediaData)
```

head() is used to return the first 5-10 entries of the dataset.

Str() is used to analyse the column datatypes and other things.

Summary() is used for advanced analysis like mean, mode, min, max, 1st Quadrant, 3rd Quadrant etc.

For some technical issue we can’t give you the output of these. Because you will know while you execute them.

Now its time for some charts. Yes!! Its time for Exploratory Data Analysis.

EDA Exploratory Data Analysis

1.Sentiment Distribution Chart

In this we are trying to analyse the sentiment distribution of our dataset based on column ***Sentiment Label***.

This Sentiment Label contains the data of reaction of the post like Positive, Negative, Neutral.

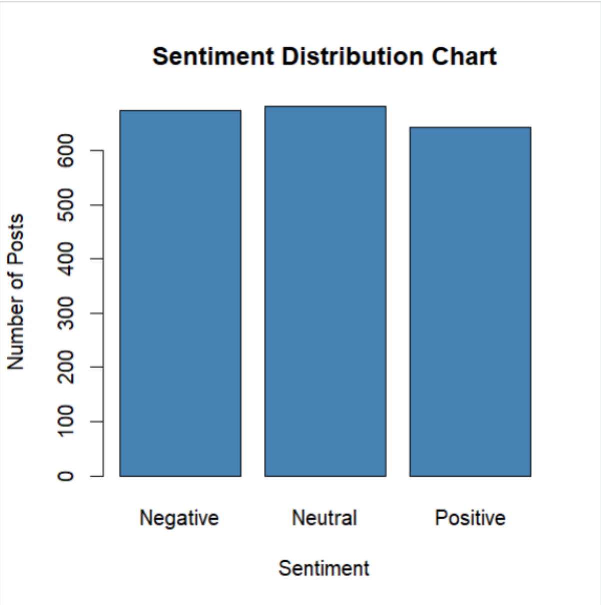
Now, we are trying to draw a chart on how many posts are Positive and how many are Negative. It is hard to count them because the dataset is huge.

We use a method called table() which counts the unique value in the column.

Then we try to draw a bar chart for the data.

```
Count_Table <-table(SocialMediaData$Sentiment.Label)

  barplot(

    Count_Table,
```

```
    col = "steelblue",
    border = "black",
    main = "Sentiment Distribution Chart",
    xlab = "Sentiment",
    ylab = "Number of Posts"
  )
```




```
Negative  Neutral Positive 
     675      682      643 
> 
```

We all know the contents in the barplot(). I don’t waste time by explain that. You will know after seeing the output. If no remove the word from your desktop. You are unworthy for this notes.

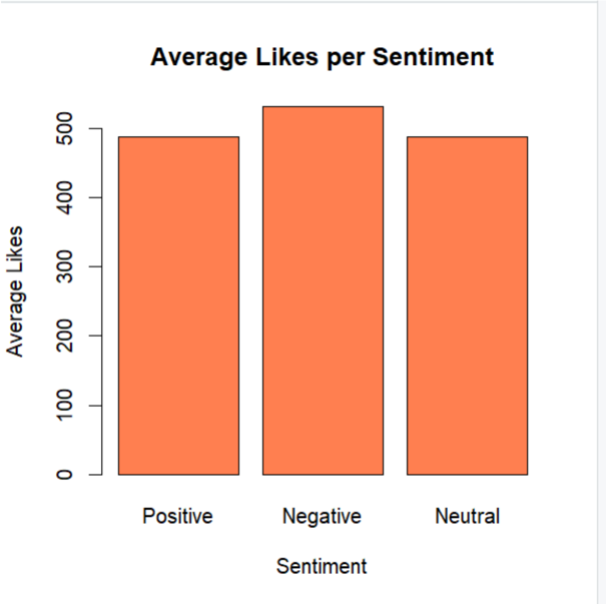
2.Likes vs Sentiment

In this we tried to plot the Average Likes for positive, negative, neutral sentiment posts.

To do this we first have to divide data based on Sentiment and sum their likes and do an average then create a data frame for average likes and try to plot the dataset as a bar chart.

```
Total_Positive_Post_Count <- sum(SocialMediaData$Sentiment.Label == "Positive")

  Total_Negative_Post_Count <- sum(SocialMediaData$Sentiment.Label == "Negative")
```

```
  Total_Neutral_Post_Count <- sum(SocialMediaData$Sentiment.Label ==
"Neutral")


  Total_Positive_Likes_Count <-
sum(SocialMediaData$Number.of.Likes[SocialMediaData$Sentiment.Label=
="Positive"])

  Total_Negative_Likes_Count <-
sum(SocialMediaData$Number.of.Likes[SocialMediaData$Sentiment.Label=
="Negative"])

  Total_Neutral_Likes_Count <-
sum(SocialMediaData$Number.of.Likes[SocialMediaData$Sentiment.Label=
="Neutral"])


  Average_Positive_Likes <- Total_Positive_Likes_Count /
Total_Positive_Post_Count

  Average_Negative_Likes <- Total_Negative_Likes_Count /
Total_Negative_Post_Count

  Average_Neutral_Likes <- Total_Neutral_Likes_Count /
Total_Neutral_Post_Count


  Average_Likes_Sentiment <- data.frame(

   Sentiment = c("Positive","Negative","Neutral"),

    Average_Likes =
c(Average_Positive_Likes,Average_Negative_Likes,Average_Neutral_Like
s)

  )

  head(Average_Likes_Sentiment)

  #2.1 Bar Plot

  barplot(

    height = Average_Likes_Sentiment $Average_Likes,

    names.arg = Average_Likes_Sentiment $Sentiment,

    col = "coral",

    border = 'black',

    xlab = "Sentiment",

    ylab = "Average Likes",

    main = "Average Likes per Sentiment"

  )
```



First, we calculated the number of Sentiment based posts in the dataset to make use of the data in calculating the average.

Then we calculated the number of likes for the Sentiment based posts and then we calculated the Average likes for Sentiment based posts.

Then we created a data frame for the average data then plotted it using barplot().

3. Most Liked Posts (Top 10 Post IDs)

To do this we can sort the data set based on likes on the posts. Then we take the head of the dataset.

So, to sort the dataset we use arrange(). Arrange() has two parameters they are as follow

1.DataSet (obviously..)

2.Column that should be sorted based on desc() / asec()

Here we need top most liked post so we use desc()

Note: arrange() method is available in dplyr library. Make sure to import the library.

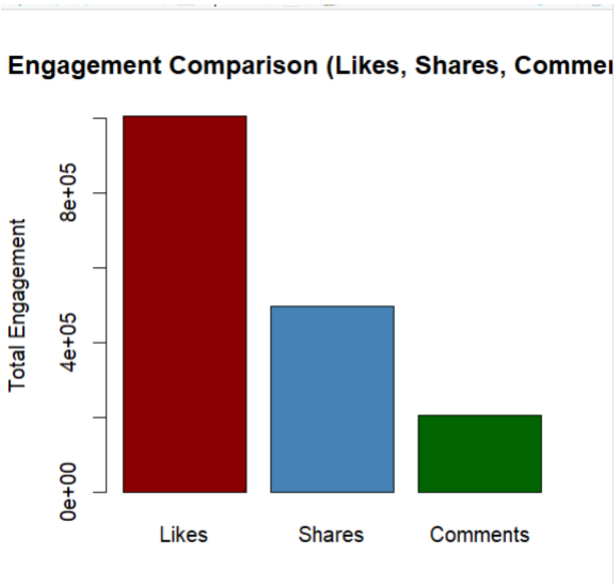
```
Sorted_by_likes <- arrange(SocialMediaData,desc(Number.of.Likes))
 head(Sorted_by_likes,10)
```

4.Engagement Comparison (Likes, Shares, Comments)

This means through which engagement feature users are more interacting with the post in social media.

To do this we first have to calculate the total number of likes, shares, comments and try to plot the dataset.

```
Engagement_score <- data.frame(
    Total_Likes = sum(SocialMediaData$Number.of.Likes),
    Total_Shares = sum(SocialMediaData$Number.of.Shares),
    Total_Comments = sum(SocialMediaData$Number.of.Comments)
  )
  engagement_values <- c(
    Engagement_score$Total_Likes,
    Engagement_score$Total_Shares,
    Engagement_score$Total_Comments
  )

  engagement_names <- c("Likes", "Shares", "Comments")

  barplot(
    height = engagement_values,
    names.arg = engagement_names,
    col = c("darkred", "steelblue", "darkgreen"),
    border = "black",
    ylab = "Total Engagement",
    main = "Engagement Comparison (Likes, Shares, Comments)"
  )
```



Through this we can confirm that users are interacting with likes than other two interacting feature.

5.Post Type vs Likes

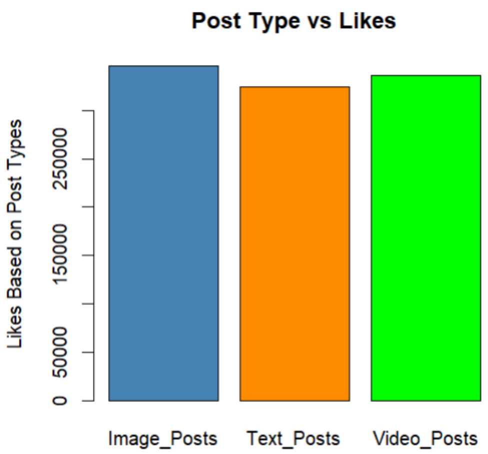
In this we are trying to which post type(image, text, video) has more likes than the other.

To do this we first have to calculate the number of likes of the posts based on the post type.

Then try to plot the dataset using barplot().

```
 Image_post_data <-
sum(SocialMediaData$Number.of.Likes[SocialMediaData$Post.Type ==
'image'])

  Text_post_data <-
sum(SocialMediaData$Number.of.Likes[SocialMediaData$Post.Type ==
'text'])

  Video_post_data <-
sum(SocialMediaData$Number.of.Likes[SocialMediaData$Post.Type ==
'video'])


  Post_based_on_likes <- c(

    Image_post_data,

    Text_post_data,

    Video_post_data
```

```
)
Post_like_names <- c("Image_Posts","Text_Posts","Video_Posts")
barplot(
  height = Post_based_on_likes,
  names.arg = Post_like_names,
  col = c("steelblue","darkorange","green"),
  border = "black",
  ylab = "Likes Based on Post Types",
  main = "Post Type vs Likes"
)
```



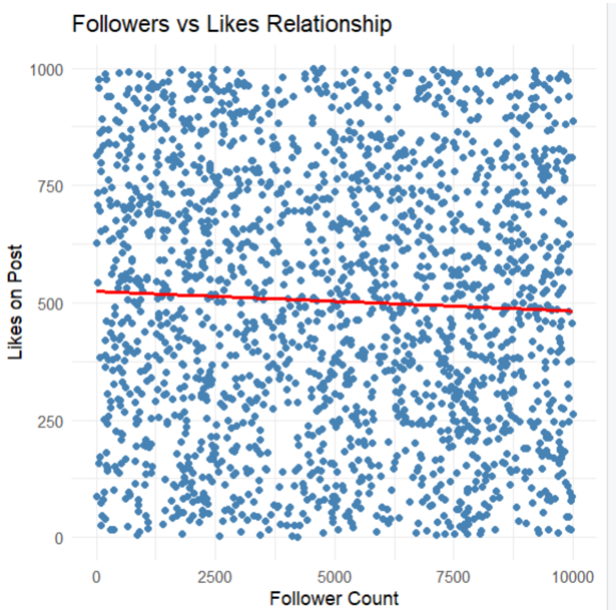
5.Followers vs Likes chart

In this we are trying to plot a chart that show the relation between the likes and the followers count for a post and admin.

For example, if 10 followers a post got only 2 likes. (simple example)

```
ggplot(
    SocialMediaData,
    aes(x = User.Follower.Count, y = Number.of.Likes)
  ) +
    geom_point(color = "steelblue") +
    geom_smooth(method = "lm", se = FALSE, color = "red") +
    labs(
      x = "Follower Count",
```

```
    y = "Likes on Post",
    title = "Followers vs Likes Relationship"
  ) +
  theme_minimal()
```



In this code geom_smooth() will draw a line between the dots using machine algorithm called logistic regression (lm). This is also called a trend line.

7. Language-wise Engagement

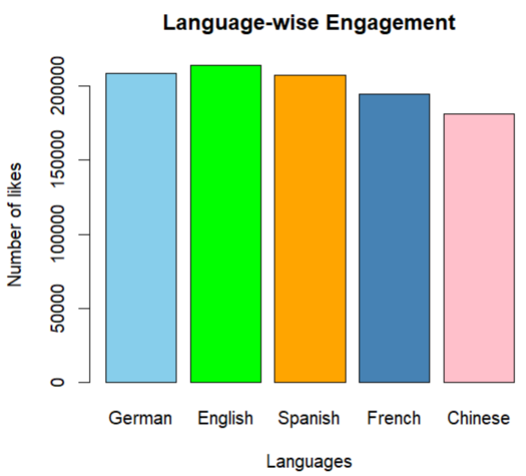
In this we try to plot the Language wise user engagement.

In this data set there are 5 languages English, German, Spanish, French, Chinese.

So, we are trying to plot total likes for a particular language.

```
Language_table <- table(SocialMediaData$Language)

  Total_de_post_Likes <-
sum(SocialMediaData$Number.of.Likes[SocialMediaData$Language ==
"de"])

  Total_en_post_Likes <-
sum(SocialMediaData$Number.of.Likes[SocialMediaData$Language ==
"en"])
```

```
  Total_es_post_Likes <-
sum(SocialMediaData$Number.of.Likes[SocialMediaData$Language ==
"es"])

  Total_fr_post_Likes <-
sum(SocialMediaData$Number.of.Likes[SocialMediaData$Language ==
"fr"])

  Total_zh_post_Likes <-
sum(SocialMediaData$Number.of.Likes[SocialMediaData$Language ==
"zh"])


  Language_based_data <- c(

    German_likes = Total_de_post_Likes,

    English_likes = Total_en_post_Likes,

    Spanish_likes = Total_es_post_Likes,

    French_likes = Total_fr_post_Likes,

    Chinese_likes = Total_zh_post_Likes

  )

  languages_cols =
c("German","English","Spanish","French","Chinese")

  barplot(

    Language_based_data,

    names.arg = languages_cols,

    col = c("skyblue","green","orange","steelblue","pink"),

    border = "black",

    xlab = "Languages",

    ylab = "Total likes",

    main = "Language-wise Engagement"

  )
```

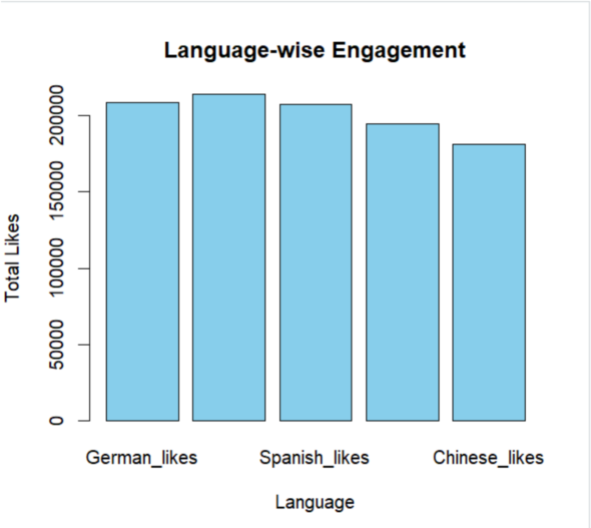


There is an easy and short way to do it using dplyr().

```
Language_based_data_short <- tapply(
  SocialMediaData$Number.of.Likes,
  SocialMediaData$Language,
  sum
)

barplot(
  Language_based_data,
  col = "skyblue",
  border = "black",
  xlab = "Language",
  ylab = "Total Likes",
  main = "Language-wise Engagement"
)
```

tapply() splits the data and groups the data and applies the aggregated function on the data.

In tapply() we have give the order of the inputs very carefully first input is the values and second input is to on which basis data should be grouping and the third one is the function.



8. Followers Based Likes

In this we are trying to plot data as if the followers counts effect the engagement (Likes).

To do this we split the data into 3 levels

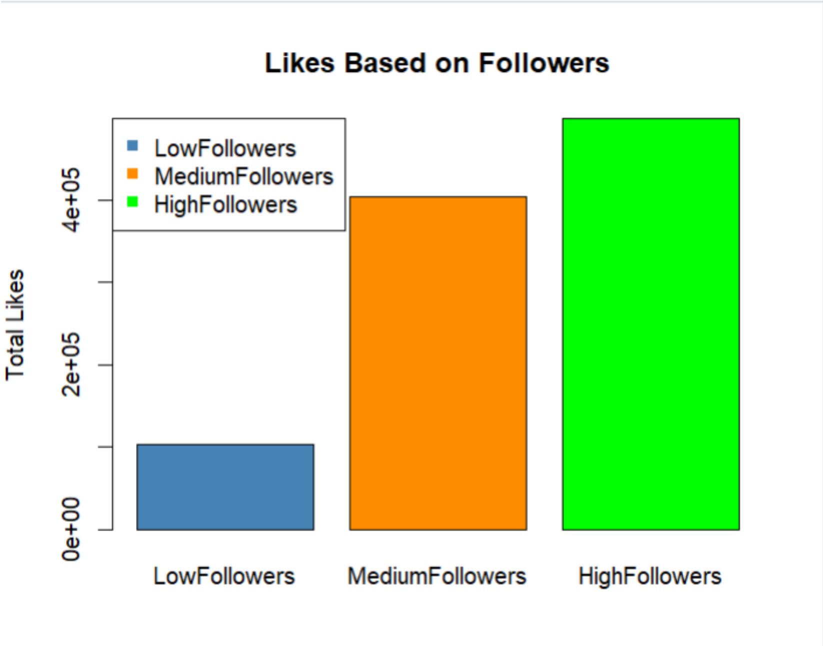
1. Low Followers (<1000)
2. Medium Follower (>=1000 and <5000)
3. High Followers (>5000)

After splitting the data set we try to plot with bar plot.

```
Low_Followers <- sum(SocialMediaData$Number.of.Likes[
    SocialMediaData$User.Follower.Count < 1000
  ])


  Medium_Followers <- sum(SocialMediaData$Number.of.Likes[
    SocialMediaData$User.Follower.Count >= 1000 &
      SocialMediaData$User.Follower.Count < 5000
  ])


  High_Followers <- sum(SocialMediaData$Number.of.Likes[
    SocialMediaData$User.Follower.Count >= 5000
  ])
```

```
  Followers_Based_likes <- c(
    Low_Followers_users = Low_Followers,
    Medium_Followers_Users = Medium_Followers,
    High_Followers_Users = High_Followers
  )
  Followers_cols =
c("LowFollowers","MediumFollowers","HighFollowers")


  barplot(
    Followers_Based_likes,
    names.arg = Followers_cols,
    col = c("steelblue","darkorange","green"),
    border = "black",
    ylab = "Total Likes",
    main = "Likes Based on Followers"
    )
  legend(
    "topleft",
    legend = Followers_cols,
    col = c("steelblue","darkorange","green"),
    pch = 15
  )
```

There is nothing new except the legend part. Legend is used to add the axis part to the graph.



In the result we clearly seen that followers count is effecting the total likes.

9. Text Preprocessing

In this we are trying to analyse the text data in the dataset. In the dataset we have text data as Post Content.

First, we convert all of the data into lower case. (easy to analyse)

Then, we removing special characters and numbers form the data.

Then, we remove the unwanted text like the, that, this etc.

At last, we are splitting the data and converting them into numeric data using TF (Term Frequency), IDF (Inverse Document Frequency).

Now, we are using corpus for text analysis.

Corpus is a collection of documents used for text analysis in NLP and ML algorithms.

We are converting the Tokenized data into corpus because ML algorithms don't directly analyse the raw text.

```
#Text Pre-processing
  #1. Converting into Lower Case
  SocialMediaData$Clean_Post_Content <-
tolower(SocialMediaData$Post.Content)
```

```
  #2. Removing numbers and Special Characters

  SocialMediaData$Clean_Post_Content <- gsub("[^a-z
]","",SocialMediaData$Clean_Post_Content)


  #3. Removing Unused words

  stops_words <- c("the","is","am","are","and","this","that")

  SocialMediaData$Clean_Post_Content <-
gsub(paste(stops_words,collapse =
"|"),"",SocialMediaData$Clean_Post_Content)


  #4. Tokenization

  SocialMediaData$Clean_Post_Content <-
strsplit(SocialMediaData$Clean_Post_Content , " ")

  head(SocialMediaData$Clean_Post_Content)


  #TF-IDF

  #TF (Term Frequency)

  #IDF (Inverse Document Frequency)


  corpus <-
VCorpus(VectorSource(SocialMediaData$Clean_Post_Content))


  tfidf_df <- DocumentTermMatrix(

    corpus,

    control = list(

      weighting = weightTfIdf

    )

  )


  dim(tfidf_df)
```

gsub() is used to remove the unwanted data. It’s like replacing the unwanted text with space.

strsplit() is used to split the data into tokens.

VCorpus() Converts the tokenized data into corpus.

VectorSource() will tell the RStudio that the input data is a Vector.

DocumentTermMatrix() converts the documents into matrix of words.

- **Rows** → Documents (social media posts)
- **Columns** → Terms (words)
- **Cells** → Weight / importance of a word in a document

TF → word importance in **one post**

IDF → word rarity across **all posts**

TF × IDF → final weight

dim() is used to know the dimension of the data set.

10. Logistic Regression

This is one the supervised learning algorithm in Machine Learning. It is used when the output is categorical type like YES/NO, SPAM/NOT SPAM.

In R programming to apply logistic regression we have a method called glm() which stands for logistic model. Our idea is to use this glm() for predicting the data of Sentiment Label (‘Positive’, ‘Neutral’, ‘Negative’)

But unfortunately, glm() method only takes binomial type of data i.e column with two unique data.

Here we have 2 options either remove one the Sentiment Label and use glm() or use multinom() for multinomial regression.

So, we used a new method called multinom() which is used for applying the logistic regression for multinomial columns.

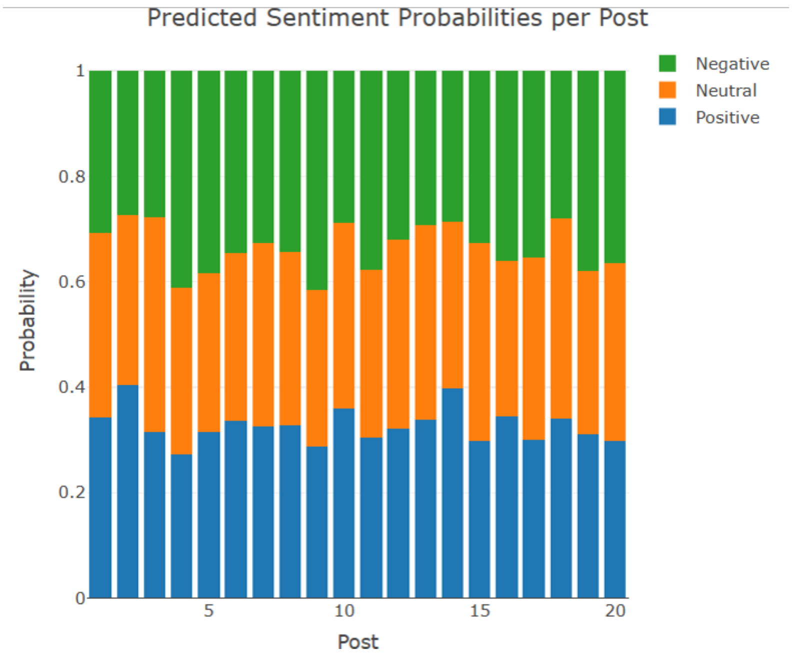
```
SocialMediaData$Sentiment.Label <-
as.factor(SocialMediaData$Sentiment.Label)


  multiclass_logistic_model <- multinom(

    Sentiment.Label ~ Number.of.Likes + Number.of.Shares +

      Number.of.Comments + User.Follower.Count,

    data = SocialMediaData

  )


  summary(multiclass_logistic_model)
```

```
Call:
multinom(formula = Sentiment.Label ~ Number.of.Likes + Number.of.Shares +
    Number.of.Comments + User.Follower.Count, data = SocialMediaData)

Coefficients:
         (Intercept) Number.of.Likes Number.of.Shares Number.of.Comments User.Follower.Count
Neutral   0.04900487   -0.0005346093    -6.935389e-05        0.001603730         1.714785e-05
Positive  0.39935092   -0.0005539718    -6.437049e-04        0.001021118        -2.237192e-05

Std. Errors:
          (Intercept) Number.of.Likes Number.of.Shares Number.of.Comments User.Follower.Count
Neutral  1.707825e-06    0.0001648108     0.0003310843       0.0008251341         1.624397e-05
Positive 1.785197e-06    0.0001663604     0.0003372381       0.0008364612         1.656837e-05

Residual Deviance: 4371.824
AIC: 4391.824
> 
```

We have analysed the data using logistic regression now we to predict the data using the predict() method.

We have bunch of data so we need to use small amount of data from the predicted data.

```
predicted_probs <- predict(multiclass_logistic_model,type = "probs")


  predicted_probs_dataframe <- as.data.frame(predicted_probs)


  predicted_probs_dataframe$Post_ID
->1:nrow(predicted_probs_dataframe)
  prob_df_sample <- predicted_probs_dataframe[1:20, ]


  plot_ly(prob_df_sample, x = ~Post_ID, y = ~Positive, type = "bar",
name = "Positive") %>%
    add_trace(y = ~Neutral, name = "Neutral") %>%
    add_trace(y = ~Negative, name = "Negative") %>%
    layout(
      barmode = "stack",
      title = "Predicted Sentiment Probabilities per Post",
      xaxis = list(title = "Post"),
      yaxis = list(title = "Probability")
    )
```



Now, we are trying to predict the Sentimental Label based on the words that are in the Post Content using Naïve Bayes Machine Learning model.

So, to use the Naïve Bayes we have a library called 'e1071'. This method takes only data frame as the input. We need to convert the tfidf data into matrix then to data frame.

Then we split the data into train data and test data in the ration 7:1. Then we use the Naïve Bayes method to analyse the data. After we got the analysed data we can use the predict() to predict the future values.

Lastly, we create a confusion matrix to visualize the data. To visualize the data we are using plot_ly().

```
tfidf_matrix <- as.matrix(tfidf_df)


  naive_bayes_data <- data.frame(
    tfidf_matrix,
    Sentiment.Label = SocialMediaData$Sentiment.Label
  )
  set.seed(18)


  split <- sample.split(naive_bayes_data$Sentiment.Label, SplitRatio
= 0.7)
  test_data <- naive_bayes_data[split == FALSE, ]
  train_data <- naive_bayes_data[split == TRUE, ]
```

```
  naive_bayes_model <- naiveBayes(Sentiment.Label ~. ,data =
train_data)

  summary(naive_bayes_model)


  naive_bayes_prediction <- predict(naive_bayes_model, newdata =
test_data)

  summary(naive_bayes_prediction)


  #Confusion Matrix

  conf_mat <- table(

    Predicted = naive_bayes_prediction,

    Actual = test_data$Sentiment.Label

  )



  summary(conf_mat)


  #Visualize the Naive Bayes Data

  confusion_matrix_data <- as.data.frame(conf_mat)


  plot_ly(

    data = confusion_matrix_data,

    x = ~Actual,

    y = ~Predicted,

    z = ~Freq,

    type = "heatmap",

    colors = colorRamp(c("skyblue", "blue"))


  ) %>%

    layout(

      title = "Naive Bayes Confusion Matrix",

      xaxis = list(title = "Actual Sentiment"),
```

```
  yaxis = list(title = "Predicted Sentiment")
)
```