VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Programming for Data Science

Digital Assignment-1

COURSE : Programming For Data Science

Semester : Winter Semester 2024-25

COURSE ID : BCSE 207L

CLASS NO. : VL2024250502043

SLOT : B2 + TB2

SCHOOL : SCOPE

FACULTY : Prof. RAJKUMAR R

SUBMITTED BY :

GANESH MAHARAJ KAMATHAM (*22BDS0168*)

Reddit Subreddit Dashboard using R Shiny

1. Introduction

With the rise of online communities, Reddit has become a crucial platform for analyzing trends, sentiments, and discussions. This project aims to develop a **Reddit Subreddit Dashboard** using **R Shiny**, allowing users to explore subreddit activity, sentiment trends, and word frequencies.

2. Objective

The goal of this project is to:

- Analyze subreddit activity and rankings.
- Perform sentiment and topic-based classification.
- Visualize word frequency, trends, and engagement metrics.
- Provide an interactive R Shiny dashboard for real-time exploration.

3. Data Collection & Preprocessing

3.1 Data Source

- The dataset was obtained from a **web-scraped Reddit dataset**, containing subreddit posts with **title, text content, score, comments, and flair**.

3.2 Data Preprocessing

- Removed missing values and duplicates.
- Converted text to lowercase and removed punctuation/stopwords.
- Created additional columns based on **sentiment categories** and **topics** using predefined word lists.

4. Methodology

4.1 Sentiment & Topic Classification

- Used **word-based classification** to assign sentiment scores (e.g., happiness, depression, anger, gratitude).
- Classified posts based on **topics** like politics, fitness, technology, etc.

4.2 Visualization Techniques

- Bar charts, line charts, word clouds, and heatmaps to present subreddit trends.
- Used plotly, ggplot2, tm, and wordcloud libraries for interactive visuals.

5. Dashboard Features & Visualizations

The dashboard consists of multiple interactive components:

- Most Active Subreddits → Bar chart showing subreddits with the most posts.
- Subreddit Rankings → Ranking of subreddits based on sentiment/topic scores.
- Flair Distribution → Pie chart showing the distribution of post flairs.
- Score Trend → Line chart visualizing how post scores change over time.
- Comment Trend → Line chart tracking comment trends across subreddits.
- Sentiment Distribution → Bar chart displaying sentiment analysis results.
- Topic Heatmap → Heatmap showing dominant topics in different subreddits.
- Word Cloud → Visualizing the most common words in a selected subreddit.
- Word Frequency → A detailed table of frequently used words.
- Data Table → Complete dataset for detailed exploration.

Implementation in R Shiny :

```
# Load dataset (ensure your CSV has 'Subreddit', 'Title', 'Text_Content' columns)
data <- read.csv("reddit_data.csv")

# Define word categories
word_categories <- list(
  happiness = c("joy", "happy", "excited", "cheerful","yay","hurray"),
  depression = c("sad", "lonely", "anxious", "depressed","alone","low"),
  anger = c("angry", "hate", "furious", "rage", "fuck", "bsdk", "lawde", "chutiya"),
  gratitude = c("thank", "grateful", "appreciate", "kind"),
  romantic = c("love", "crush", "date", "relationship", "sex", "beautiful", "boyfriend","girlfriend"),
  religious = c("god", "church", "prayer", "faith",
"muslim","christian","Hindu","sikh","jain","Buddhist"),
  political = c("election", "government", "bjp", "congress", "vote", "dmk", "admk",
"IND","TDP","YSRCP","IUML"),
  tech = c("ai", "coding", "startup", "technology","gpt","developer","money"),
  foodie = c("food", "restaurant", "cooking", "biryani", "chai","chicken", "fish", "mutton"),
  fitness = c("gym", "workout", "running", "calories", "exercise","military","marathon"),
```

```
  formal = c("sir", "respectfully", "regards",
"namasthe","namaste","vanakkam","namaskaram","namaskar"),
  informal = c("dude", "bro", "lol", "chill","yo","guys","chick"),
  literary = c("complex", "philosophy", "literature", "poetry","writing","novel"),
  meme = c("meme", "lol", "lmao", "troll","lol","hehe","funny","ngl","irl"),
  questions = c("how", "why", "what", "help","where","when","is"),
  complaints = c("bad", "problem", "issue","terrible","horrible"),
  career = c("job", "interview", "career",
"salary","money","business","startup","founder","company"),
  travel = c("trip", "travel", "vacation","long","tour","flight","train"),
  debate = c("disagree", "argue", "vs","fight",""),
  crypto = c("bitcoin", "stocks",
"trading","eth","crypto","etherium","tokens","USD","block","chain","blockchain"),
  superstition = c("astrology", "ghost", "superstition","bhoot","kalajadu","blackmagic","religion")
)

# Function to count word occurrences in a text
count_words <- function(text, words) {
  sum(str_count(tolower(text), paste(words, collapse = "|")))
}

# Apply word counting to the dataset
for (category in names(word_categories)) {
  data[[category]] <- apply(data, 1, function(row) count_words(paste(row["Title"],
row["Text_Content"]), word_categories[[category]]))
}

# UI
ui <- fluidPage(
  titlePanel("Reddit Subreddit Dashboard"),

  sidebarLayout(
```

```
    sidebarPanel(
      selectInput("selectedCategory", "Select Category", choices = names(word_categories)),
      selectInput("selectedSubreddit", "Select Subreddit for Word Cloud", choices =
unique(data$Subreddit))
    ),

    mainPanel(
      tabsetPanel(
        tabPanel("Score Trend", plotlyOutput("scoreTrend")),
        tabPanel("Comment Trend", plotlyOutput("commentTrend")),
        tabPanel("Flair Distribution", plotlyOutput("flairPie")),

        tabPanel("Subreddit Rankings", plotlyOutput("subredditPlot")),
        tabPanel("Word Cloud", plotOutput("wordCloud")),
        tabPanel("Most Active Subreddits", plotlyOutput("activeSubredditPlot")),
        tabPanel("Sentiment Distribution", plotlyOutput("sentimentPlot")),
        tabPanel("Topic Heatmap", plotlyOutput("heatmapPlot")),
        tabPanel("Word Frequency", plotlyOutput("wordFreqPlot")),
        tabPanel("Data Table", DTOutput("dataTable"))
      )
    )
  )
)

# Server
server <- function(input, output) {

  # Most Active Subreddits (based on number of posts)
  output$activeSubredditPlot <- renderPlotly({
    data %>%
      count(Subreddit, sort = TRUE) %>%
```

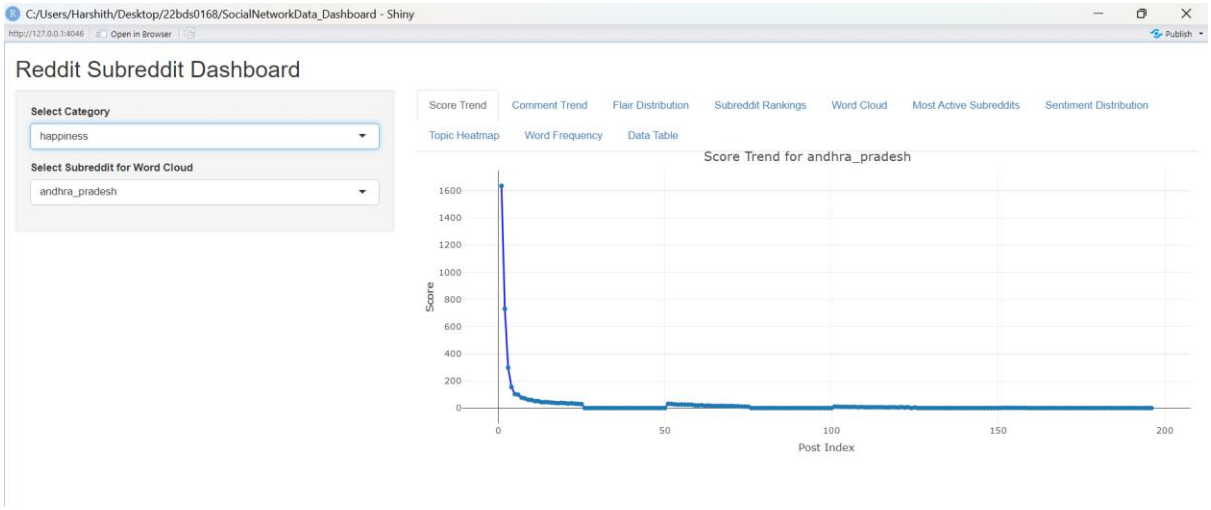
```
    head(10) %>%
    plot_ly(x = ~Subreddit, y = ~n, type = "bar", name = "Post Count") %>%
    layout(title = "Most Active Subreddits")
})

# Subreddit Ranking Plot
output$subredditPlot <- renderPlotly({
  data %>%
    group_by(Subreddit) %>%
    summarise(score = sum(.data[[input$selectedCategory]], na.rm = TRUE)) %>%
    arrange(desc(score)) %>%
    head(10) %>%
    plot_ly(x = ~Subreddit, y = ~score, type = "bar", name = input$selectedCategory) %>%
    layout(title = paste("Top Subreddits by", input$selectedCategory))
})

# Flair Distribution Pie Chart
output$flairPie <- renderPlotly({
  subset_data <- data %>% filter(Subreddit == input$selectedSubreddit) %>%
    count(Flair)

  plot_ly(subset_data, labels = ~Flair, values = ~n, type = "pie", textinfo = "label+percent",
          marker = list(colors = brewer.pal(8, "Set3"))) %>%
    layout(title = paste("Flair Distribution in", input$selectedSubreddit))
})

# Score Trend Line Chart (Using Row Index)
output$scoreTrend <- renderPlotly({
  subset_data <- data %>% filter(Subreddit == input$selectedSubreddit)

  plot_ly(subset_data, x = ~seq_along(Score), y = ~Score, type = "scatter", mode = "lines+markers",
```

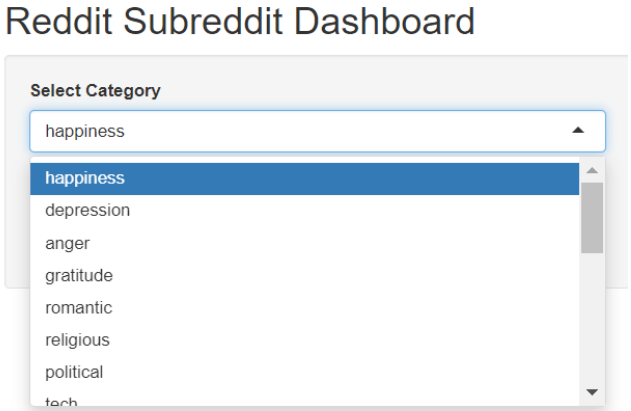
```
        line = list(color = "blue")) %>%
    layout(title = paste("Score Trend for", input$selectedSubreddit),
           xaxis = list(title = "Post Index"),
           yaxis = list(title = "Score"))
 })

 # Comment Trend Line Chart (Using Row Index)
 output$commentTrend <- renderPlotly({
  subset_data <- data %>% filter(Subreddit == input$selectedSubreddit)

  plot_ly(subset_data, x = ~seq_along(Comments), y = ~Comments, type = "scatter", mode =
"lines+markers",
          line = list(color = "red")) %>%
    layout(title = paste("Comment Trend for", input$selectedSubreddit),
           xaxis = list(title = "Post Index"),
           yaxis = list(title = "Comments"))
 })

 # Sentiment Distribution Per Subreddit
 output$sentimentPlot <- renderPlotly({
  data %>%
    group_by(Subreddit) %>%
    summarise(Happiness = sum(happiness),
              Depression = sum(depression),
              Anger = sum(anger),
              Gratitude = sum(gratitude)) %>%
    gather(key = "Sentiment", value = "Count", -Subreddit) %>%
    arrange(desc(Count)) %>%
    head(40) %>%
    plot_ly(x = ~Subreddit, y = ~Count, color = ~Sentiment, type = "bar") %>%
    layout(title = "Sentiment Distribution Across Subreddits")
```

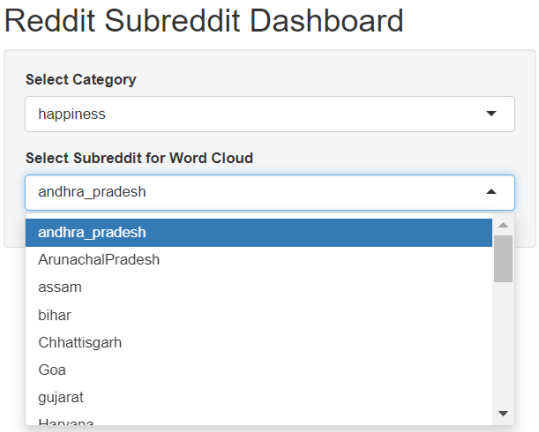
```
  })

  # Topic-Based Heatmap
  output$heatmapPlot <- renderPlotly({
    heatmap_data <- data %>%
      group_by(Subreddit) %>%
      summarise(Tech = sum(tech),
                Political = sum(political),
                Foodie = sum(foodie),
                Fitness = sum(fitness)) %>%
      gather(key = "Topic", value = "Count", -Subreddit)

    plot_ly(heatmap_data, x = ~Subreddit, y = ~Topic, z = ~Count, type = "heatmap", colors = "Blues") %>%
      layout(title = "Topic Intensity Heatmap")
  })



  # Word Cloud for Selected Subreddit
  output$wordCloud <- renderPlot({
    subset_data <- data %>% filter(Subreddit == input$selectedSubreddit)
    text_corpus <- Corpus(VectorSource(paste(subset_data$Title, subset_data$Text_Content)))
    text_corpus <- tm_map(text_corpus, content_transformer(tolower))
    text_corpus <- tm_map(text_corpus, removePunctuation)
    text_corpus <- tm_map(text_corpus, removeWords, c(stopwords("english"), "text"))  # Remove "text"

    wordcloud(text_corpus, max.words = 100, random.order = FALSE, colors = brewer.pal(8, "Dark2"))
  })
```

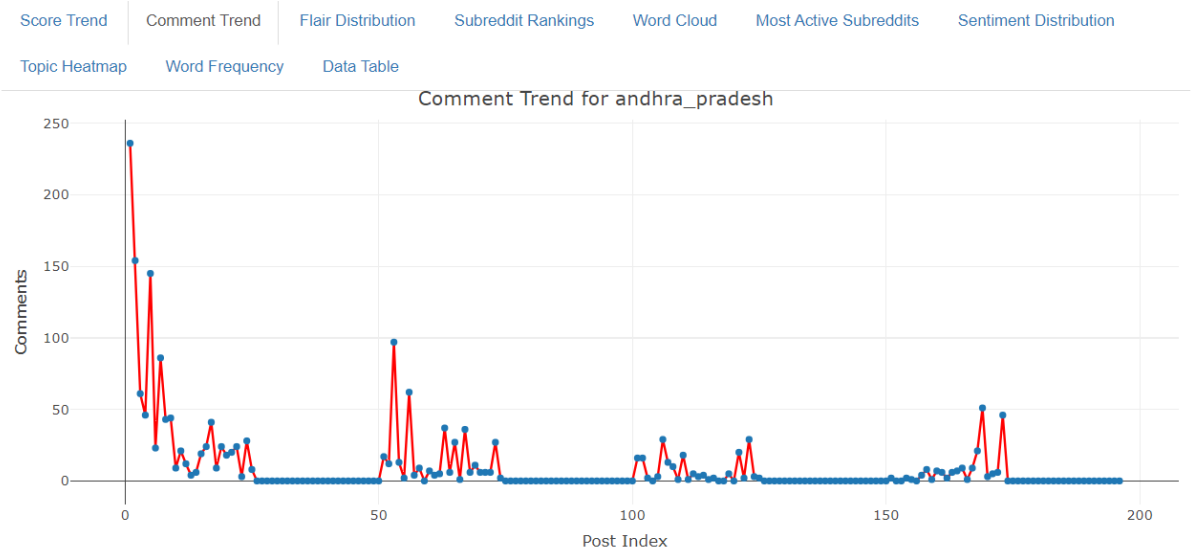
```
  # Word Frequency Bar Chart for Selected Subreddit
  output$wordFreqPlot <- renderPlotly({
    subset_data <- data %>% filter(Subreddit == input$selectedSubreddit)
    words <- unlist(strsplit(tolower(paste(subset_data$Title, subset_data$Text_Content)), "\\s+"))
    words <- words[!words %in% stopwords("english") & nchar(words) > 3]
    word_counts <- as.data.frame(table(words)) %>% arrange(desc(Freq)) %>% head(10)

    plot_ly(word_counts, x = ~words, y = ~Freq, type = "bar", name = "Word Frequency") %>%
      layout(title = paste("Most Common Words in", input$selectedSubreddit))
  })

  # Data Table
  output$dataTable <- renderDT({
    datatable(data)
  })
}

# Run the app
shinyApp(ui = ui, server = server)
```

DASHBOARD :

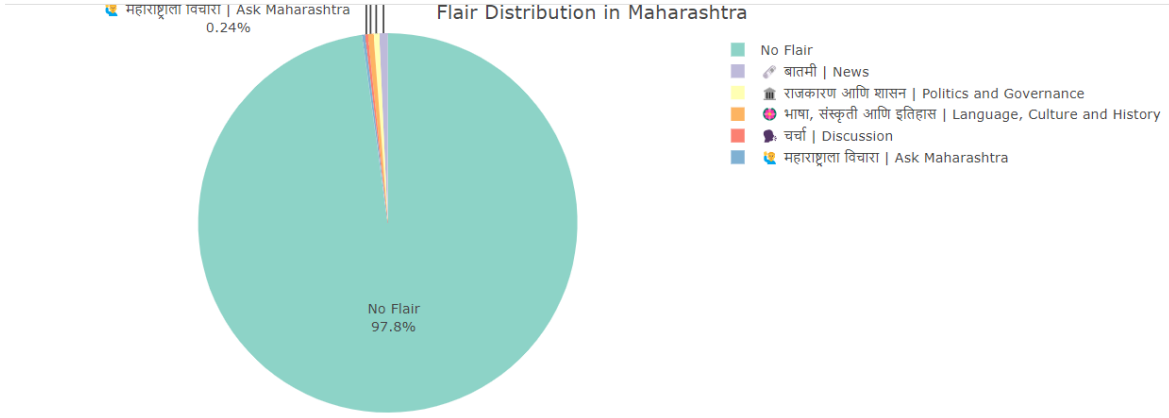

C:/Users/Harshith/Desktop/22bds0168/SocialNetworkData_Dashboard - Shiny
Reddit Subreddit Dashboard
Select Category
happiness
Select Subreddit for Word Cloud
andhra_pradesh
Score Trend
Comment Trend
Flair Distribution
Subreddit Rankings
Word Cloud
Most Active Subreddits
Sentiment Distribution
Topic Heatmap
Word Frequency
Data Table
Score Trend for andhra_pradesh
Score
Post Index


Reddit Subreddit Dashboard
Select Category
happiness
happiness
depression
anger
gratitude
romantic
religious
political
tech

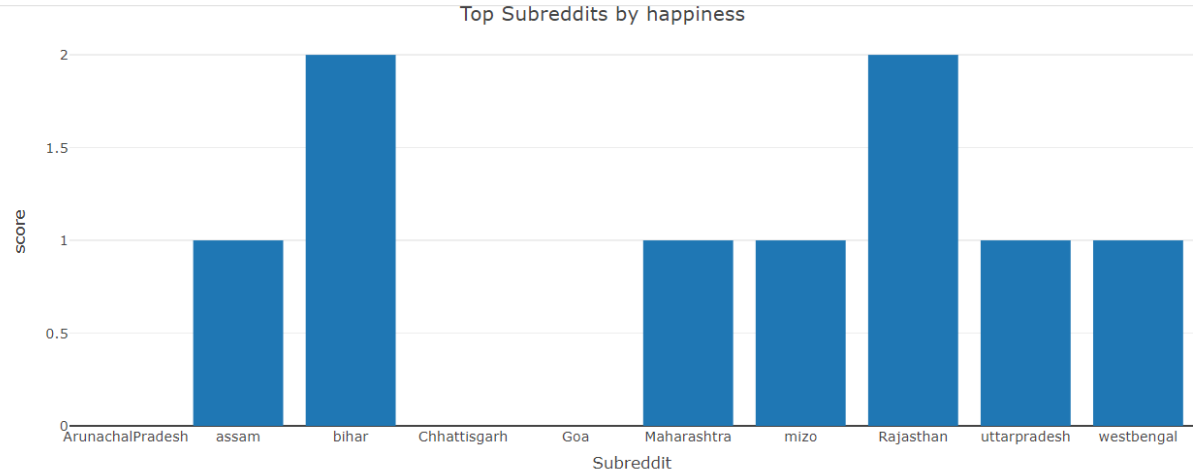

Reddit Subreddit Dashboard
Select Category
happiness
Select Subreddit for Word Cloud
andhra_pradesh
andhra_pradesh
ArunachalPradesh
assam
bihar
Chhattisgarh
Goa
gujarat


Score Trend
Comment Trend
Flair Distribution
Subreddit Rankings
Word Cloud
Most Active Subreddits
Sentiment Distribution
Topic Heatmap
Word Frequency
Data Table
Comment Trend for andhra_pradesh
Comments
Post Index

Score Trend | Comment Trend | Flair Distribution | Subreddit Rankings | Word Cloud | Most Active Subreddits | Sentiment Distribution | Topic Heatmap | Word Frequency | Data Table

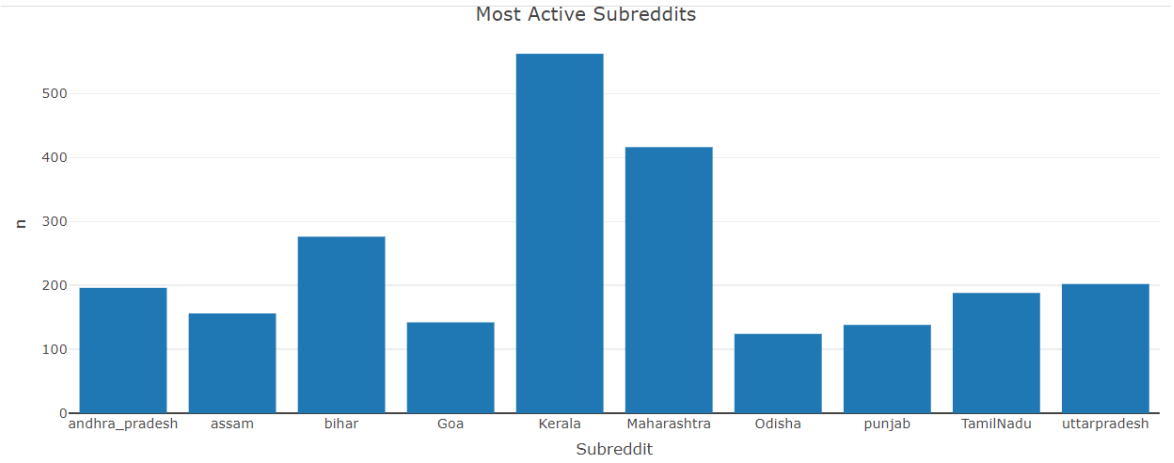


Score Trend | Comment Trend | Flair Distribution | Subreddit Rankings | Word Cloud | Most Active Subreddits | Sentiment Distribution | Topic Heatmap | Word Frequency | Data Table

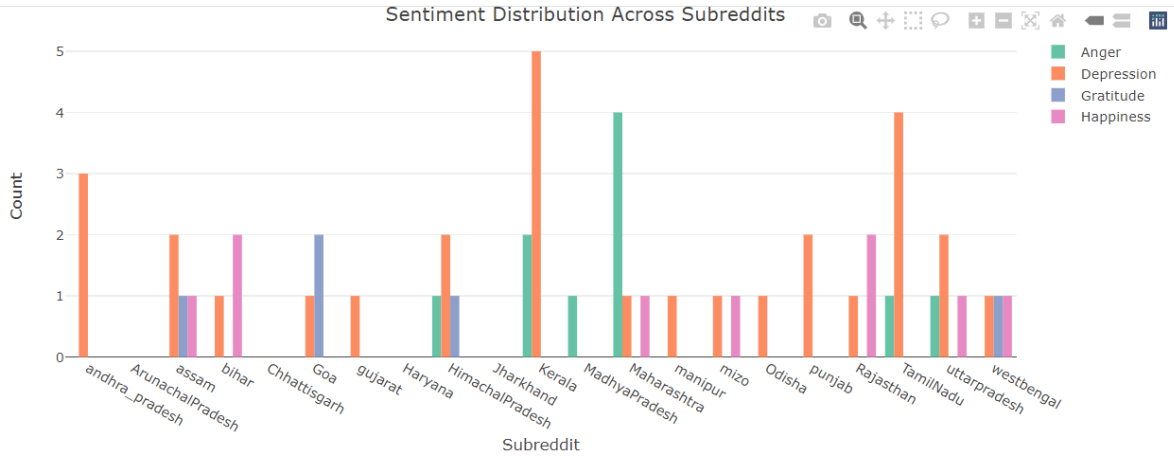


Score Trend | Comment Trend | Flair Distribution | Subreddit Rankings | Word Cloud | Most Active Subreddits | Sentiment Distribution | Topic Heatmap | Word Frequency | Data Table

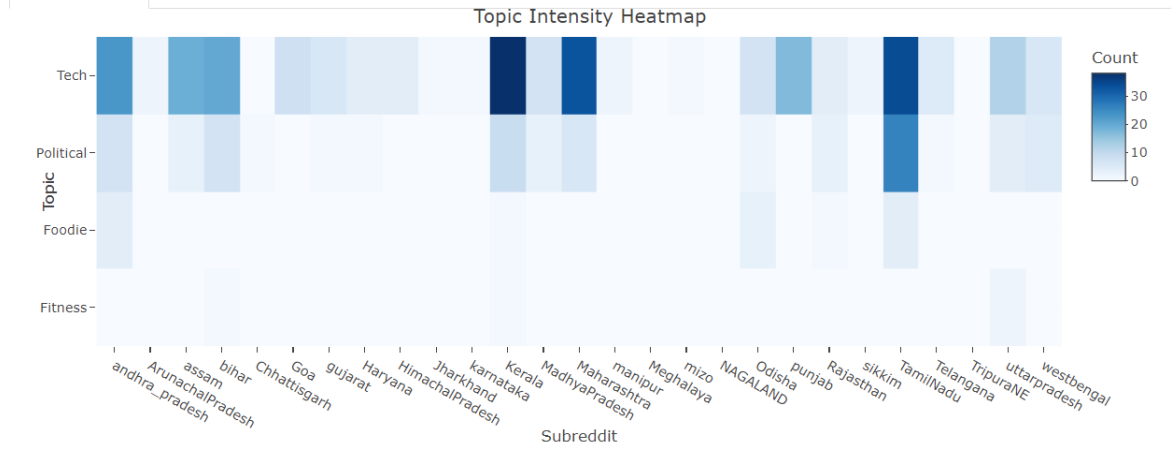
Score Trend | Comment Trend | Flair Distribution | Subreddit Rankings | Word Cloud | Most Active Subreddits | Sentiment Distribution | Topic Heatmap | Word Frequency | Data Table



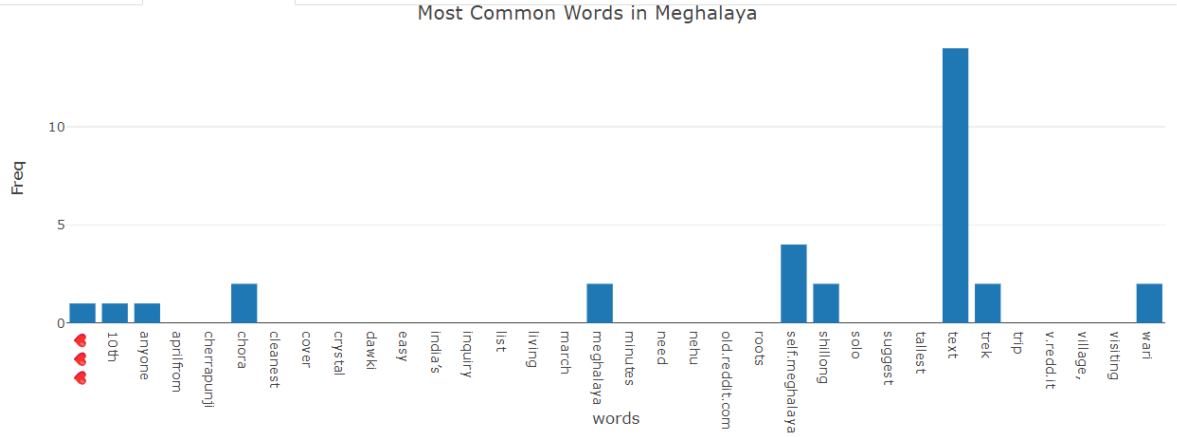
Score Trend | Comment Trend | Flair Distribution | Subreddit Rankings | Word Cloud | Most Active Subreddits | Sentiment Distribution | Topic Heatmap | Word Frequency | Data Table



Score Trend | Comment Trend | Flair Distribution | Subreddit Rankings | Word Cloud | Most Active Subreddits | Sentiment Distribution | Topic Heatmap | Word Frequency | Data Table



Score Trend | Comment Trend | Flair Distribution | Subreddit Rankings | Word Cloud | Most Active Subreddits | Sentiment Distribution | Topic Heatmap | Word Frequency | Data Table



Score Trend | Comment Trend | Flair Distribution | Subreddit Rankings | Word Cloud | Most Active Subreddits | Sentiment Distribution | Topic Heatmap | Word Frequency | Data Table

Show 10 entries Search:

	Title	Link	Author	Score	Comments	Post_Time
11	Andhra weans away Rs 9,700 cr investments in 3 months from other Southern states	https://economictimes.indiatimes.com/news/india/andhra-attracts-rs-9700-cr-investments-in-3-months/articleshow/119440678.cms	spiritgod00	51	21	Not Availabl
12	economictimes.indiatimes.com	/domain/economictimes.indiatimes.com/	Cal_Aesthetics_Club	51	12	Not Availabl
13	Remember NARA CHANDRABABU NAIDU, you didn't even build 2 government medical colleges in your 15 year CM career. Don't joke about launching satellites	https://i.redd.it/5l57bw03a3re1.jpeg	Fabulous-Fun-1628	43	4	Not Availabl
14	i.redd.it	/domain/i.redd.it/	Cal_Aesthetics_Club	44	6	Not Availabl
15	What is going on at Tirumala?	https://i.redd.it/ix5xfbgox3re1.jpeg	Fabulous-Fun-1628	43	19	Not Availabl
16	i.redd.it	/domain/i.redd.it/	Witty_Win_4872	40	24	Not Availabl
17	As ANDHRA youth is earning 10 lakhs per month working from home in TDP government, producers are	https://i.redd.it/7vpcjbvl9xqe1.jpeg	ThelaSonas	38	41	Not Availabl