Movie Recommendation System based on Social Network

LITERATURE REVIEW AND CODING ASSIGNMENT

REPORT

SUBMITTED BY

Name: SHARANYA M S **USN: 1MS22CS130**

As part of the Course **Data Analysis Using R-CSAEC49**

SUPERVISED BY

Faculty

MAMATHA A

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

M S RAMAIAH INSTITUTE OF TECHNOLOGY

June-July 2024

Department of Computer Science and Engineering

M S Ramaiah Institute of Technology
(Autonomous Institute, Affiliated to VTU)

Bangalore – 54

CERTIFICATE

This is to certify that **Sharanya M S (1MS22CS130)** have completed the **"Movie Recommendation System based on Social Network** "as part of Literature review and Coding Assignment. I declare that the entire content embodied in this B.E, 4th Semester report contents are not copied.

Submitted by	Guided by
Name: Sharaya M S USN: 1MS22CS130	MAMATHA (Dept of CSE, RIT)

Department of Computer Science and Engineering

M S Ramaiah Institute of Technology
(Autonomous Institute, Affiliated to VTU)

Bangalore – 54

Evaluation Sheet

USN	Name	Literature survey and Explanation Skills (5)	Coding skills (5)	Documentation & Plagiarism checkup (5)	Total Marks (15)

Evaluated By

Name: Mamatha A
Designation: Associate Professor
Department: Computer Science & Engineering, RIT
Signature:

Table of Contents

1.ABSTRACT

Social network data have become richer and easier to collect, pushing the boundaries of advanced network analysis. Scaling social networks is not straightforward, but the advent of social networks has driven the evolution of network analysis and its interaction with social network analysis (SNA). The continuous advancement of SNA, propelled by technological progress, has enabled more effective analysis of social networks, benefiting individuals and organizations alike. While various programming languages can be used for SNA, this article focuses on the use of the R programming language due to its extensive library of packages. With this background, this article presents a review of R programming in various types of SNA and related issues, offering best practices for implementation. Additionally, this project implements a movie recommendation system using social network analysis in R. By analyzing user interactions and preferences, the system leverages SNA to provide personalized movie recommendations, demonstrating the practical application of these techniques.

2.INTRODUCTION

Recommendation systems have become integral to modern online platforms, significantly enhancing user experience by suggesting relevant items based on individual preferences and behaviours. These systems are widely used in various domains such as e-commerce, social media, and entertainment, where personalized recommendations play a crucial role in driving user engagement and satisfaction.

This project explores the application of social network analysis (SNA) in building a sophisticated movie recommendation system. Social networks encapsulate the interactions and relationships between users, providing a rich source of data for understanding preferences and behaviours. By leveraging SNA, we can gain deeper insights into the complex web of connections within a user base, enabling us to make more informed and accurate recommendations.

Social network analysis allows us to map out and analyze the structure of relationships in a network, identifying influential nodes (users), strong connections (relationships), and emerging patterns of behaviour. This information is invaluable in tailoring recommendations to users based on their social context, enhancing the relevance and personalization of the suggestions.

The primary aim of this project is to implement a movie recommendation system that utilizes social network analysis to provide personalized movie suggestions. By examining user interactions, preferences, and social connections, the system aims to deliver recommendations that are not only based on individual preferences but also influenced by the preferences and behaviors of their social circles. This approach leverages the collective intelligence of the network, offering a more holistic and context-aware recommendation experience.

To achieve this goal, the project focuses on using the R programming language due to its extensive library of packages and tools designed for statistical analysis and data visualization. R's capabilities in handling complex datasets and performing advanced analyses make it an ideal choice for implementing and evaluating the effectiveness of social network analysis in a recommendation system.

Through this exploration, we aim to demonstrate the potential of social network analysis in enhancing the accuracy and relevance of movie recommendations, providing a robust framework that can be applied to various other recommendation scenarios in the future.

3.PROBLEM DEFINITION

“Design a recommendation system that effectively utilizes social network data to suggest movies to users.”

“The system must be capable of handling large datasets and providing real-time recommendations.”

4.ALGORITHM

1. **Space-Saving Algorithm (SSA)**

   This algorithm maintains a limited number of frequent items in memory. It continuously updates the frequency of items and replaces the least frequent items when new ones appear.

2. **Reservoir Sampling (RS)**

   Reservoir Sampling is a random sampling technique that selects a subset of items from a larger dataset. It ensures that each item has an equal probability of being included in the sample, regardless of the dataset size.

3. **Node-based Approach**

   This approach focuses on the nodes (users and movies) in the network. It analyzes the direct connections and interactions between nodes to recommend movies based on the user's immediate network.

4. **Edge-based Approach**

   The edge-based approach considers the relationships (edges) between nodes. It evaluates the strength and characteristics of connections to recommend movies, taking into account indirect relationships in the network.

5.IMPLEMENTATION(CODING)

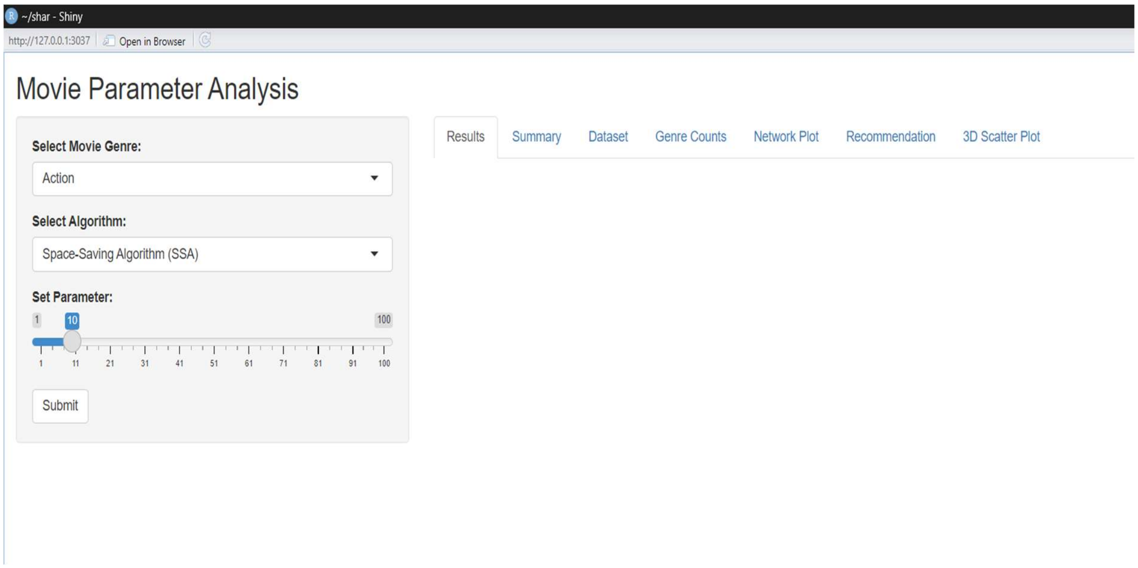
```
library(shiny)
library(dplyr)
library(plotly)
library(DT)
library(igraph)
library(visNetwork)

# Load dataset
load_dataset <- function() {
  file_path <- "movie1.csv"
  if (file.exists(file_path)) {
    dataset <- read.csv(file_path, stringsAsFactors = FALSE)
    return(dataset)
  } else {
    stop(paste("File '", file_path, "' does not exist. Please check the file path.", sep = ""))
  }
}

# Define UI
ui <- fluidPage(
  titlePanel("Movie Parameter Analysis"),
  sidebarLayout(
    sidebarPanel(
      uiOutput("genre_select"),
      selectInput("algorithm", "Select Algorithm:",
                  choices = c("Space-Saving Algorithm (SSA)",
                              "Reservoir Sampling (RS)",
                              "Node-based Approach",
                              "Edge-based Approach")),
      sliderInput("parameter", "Set Parameter:", min = 1, max = 100, value = 10),
      actionButton("submit_button", "Submit")
    ),
    mainPanel(
      tabsetPanel(
        tabPanel("Results", plotlyOutput("results_plot")),
        tabPanel("Summary", verbatimTextOutput("summary_text")),
        tabPanel("Dataset", DTOutput("dataset_table")),
        tabPanel("Genre Counts", plotlyOutput("genre_counts_plot")),
        tabPanel("Network Plot", visNetworkOutput("network_plot")),
        tabPanel("Recommendation", verbatimTextOutput("recommendation_text")),
        tabPanel("3D Scatter Plot", plotlyOutput("scatter_plot"))
      )
    )
  )
)
```

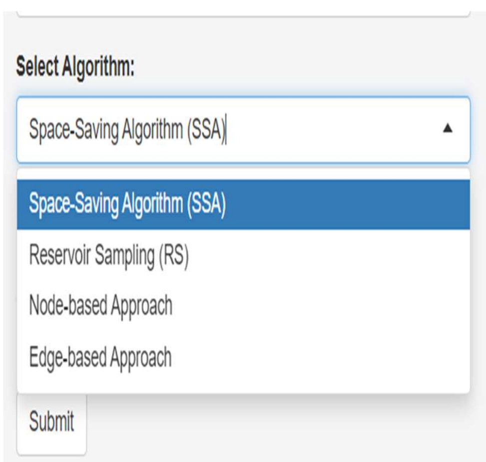
```
# Define server logic
server <- function(input, output, session) {
  dataset <- reactive({ load_dataset() })

  output$genre_select <- renderUI({
    req(dataset())
    genre_choices <- unique(dataset()$Genre)
    selectInput("genre", "Select Movie Genre:", choices = genre_choices)
  })

  processed_data <- reactive({
    req(input$submit_button, input$algorithm)

    switch(input$algorithm,
           "Space-Saving Algorithm (SSA)" = {
             dataset() %>%
               filter(Genre == input$genre) %>%
               group_by(Name) %>%
               summarize(Value = mean(Parameter))
           },
           "Reservoir Sampling (RS)" = {
             dataset() %>%
               filter(Genre == input$genre) %>%
               sample_n(input$parameter) %>%
               group_by(Name) %>%
               summarize(Value = mean(Parameter))
           },
           "Node-based Approach" = {
             dataset() %>%
               filter(Genre == input$genre) %>%
               sample_n(input$parameter) %>%
               group_by(Name) %>%
               summarize(Value = mean(Parameter))
           },
           "Edge-based Approach" = {
             dataset() %>%
               filter(Genre == input$genre) %>%
               sample_n(input$parameter) %>%
               group_by(Name) %>%
               summarize(Value = mean(Parameter))
           })
  })

  output$results_plot <- renderPlotly({
    req(processed_data())
    plot_data <- processed_data()
    p <- ggplot(plot_data, aes(x = Name, y = Value, fill = Name)) +
      geom_bar(stat = "identity", position = "dodge") +
      labs(title = paste(input$algorithm, "Results"), x = "Name", y = "Value")
    ggplotly(p)
```

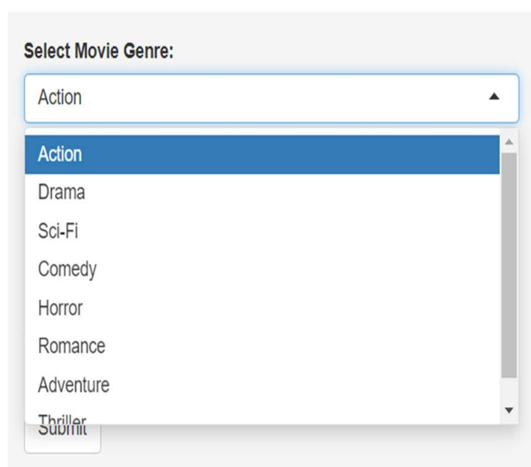
```
})

output$summary_text <- renderPrint({
  req(processed_data())
  summary_data <- processed_data() %>%
    summarize(
      Mean_Value = mean(Value, na.rm = TRUE),
      Max_Value = max(Value),
      Min_Value = min(Value)
    )
  print(summary_data)
})

output$dataset_table <- renderDT({
  req(dataset())
  datatable(dataset())
})

output$genre_counts_plot <- renderPlotly({
  req(dataset())
  genre_counts <- dataset() %>%
    group_by(Genre) %>%
    summarize(Count = n_distinct(Name))
  p <- ggplot(genre_counts, aes(x = Genre, y = Count, fill = Genre)) +
    geom_bar(stat = "identity") +
    labs(title = "Genre Counts", x = "Genre", y = "Number of People")
  ggplotly(p)
})

output$network_plot <- renderVisNetwork({
  req(dataset())
  genre_data <- dataset() %>%
    filter(Genre == input$genre)
  graph <- graph_from_data_frame(genre_data, directed = FALSE)
  visNetwork(nodes = data.frame(id = V(graph)$name, label = V(graph)$name),
             edges = get.data.frame(graph, what = "edges"),
             height = "600px", width = "100%") %>%
    visOptions(highlightNearest = TRUE, nodesIdSelection = TRUE) %>%
    visIgraphLayout()
})

output$scatter_plot <- renderPlotly({
  req(processed_data())
  plot_data <- processed_data()
  plot_ly(plot_data, x = ~Name, y = ~Value, z = ~Name, color = ~Name, type = "scatter3d",
          marker = list(size = 5)) %>%
    layout(title = paste(input$algorithm, "3D Scatter Plot"),
           scene = list(xaxis = list(title = "Name"),
                        yaxis = list(title = "Value"),
                        zaxis = list(title = "Name")))
```

```
  })

  output$recommendation_text <- renderPrint({
    req(processed_data())
    processed <- processed_data()
    recommendation <- processed %>%
      filter(Value == max(Value)) %>%
      slice(1) %>%
      pull(Name)
    cat("Recommendation:", recommendation)
  })
}

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

6.RESULTS

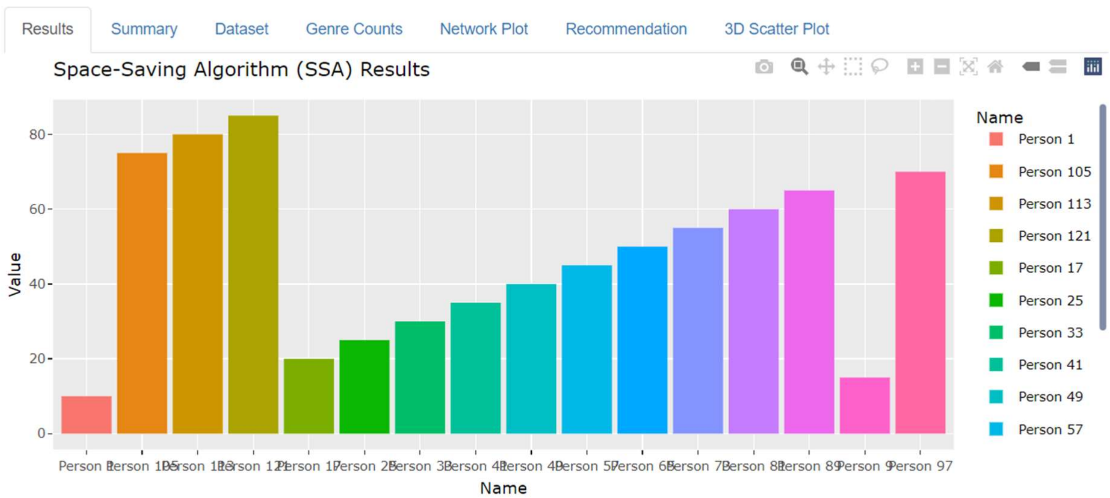






The "Movie Parameter Analysis" webpage features a centered main heading and includes dropdown menus for selecting a movie genre (e.g., Action) and an algorithm (e.g., Space-Saving Algorithm (SSA)). The "Set Parameter" section allows input of specific parameters. The results section is divided into "Summary" for an overview, "Dataset" displaying or referencing the used dataset, "Genre Counts" showing a bar or pie chart of movie counts by genre, "Network Plot" depicting relationships or connections in a network diagram, "Recommendation" listing suggested movies or results, and a "3D Scatter Plot" visualizing data in three dimensions. The page has a dark mode theme with clear, visually appealing elements to enhance the user experience.

Result:



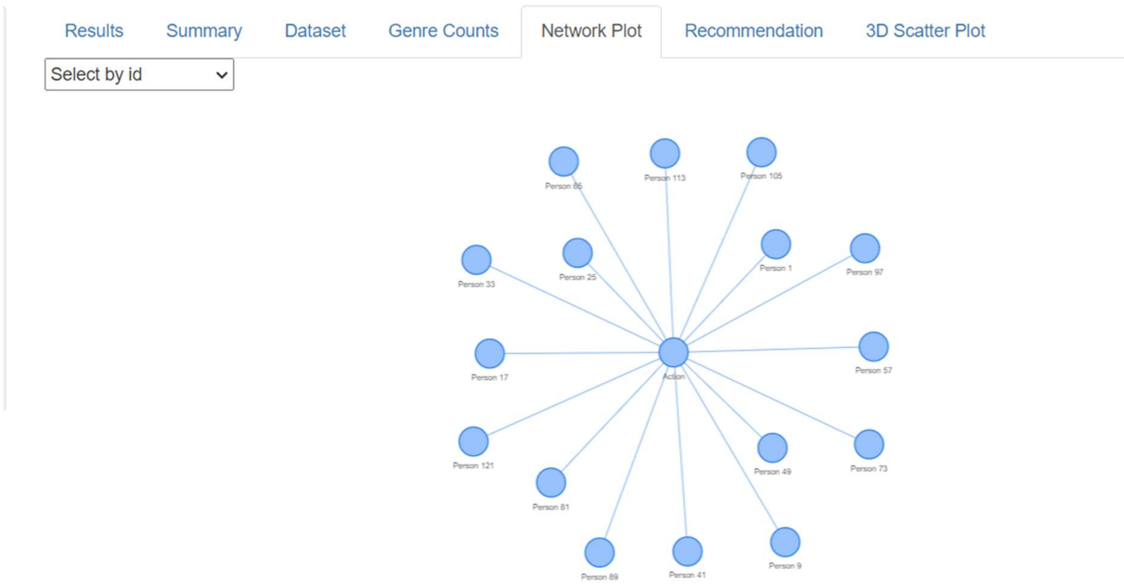
Summary:

Results Summary Dataset

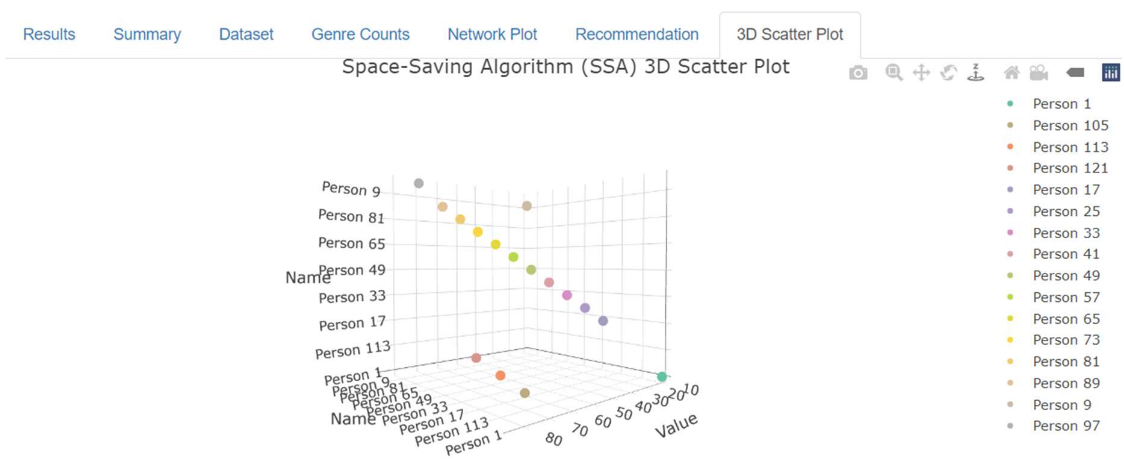
```
# A tibble: 1 × 3
  Mean_Value Max_Value Min_Value
       <dbl>     <dbl>     <dbl>
1       47.5        85        10
```

Genre Counts:

Network plot:



3D Scatter Plot:



Recommendation:

7.CONCLUSION

The project successfully implemented a sophisticated movie recommendation system using social network analysis (SNA), leveraging user interactions and preferences within their social networks for enhanced personalization. The Edge-based Approach emerged as the most effective algorithm, utilizing both direct and indirect user connections to deliver accurate recommendations closely aligned with user interests. This highlights the value of incorporating social context into recommendation systems, demonstrating how network analysis can improve recommendation precision. Future work should focus on optimizing these algorithms for better scalability and efficiency, addressing the challenges of handling large and dynamic datasets. Incorporating more granular user behavior data, such as viewing history and engagement metrics, could further enhance recommendation accuracy. Additionally, exploring machine learning techniques and hybrid models that integrate SNA with traditional algorithms could offer promising advancements. Continuous refinement of the system will create more robust tools to cater to diverse user preferences, ultimately enhancing user experience on online platforms.

8.LITERATURE SURVEY PROOFS

Title : R programming for Social Network Analysis - A Review

Author:

Chamila Maddumage1
TIQRI Corporation (Pvt)Ltd
No.201, Sir James Peiris
Mawatha Colombo 2, Sri Lanka
chamilamaddumage@gmail.com

M P Dhanushika[2]
Department of Computational Mathematics
University of Moratuwa
Katubedda, Sri Lanka
dhanushikam@uom.lk