**7-2 Final Project: Machine Learning Scenario Report**

**I. Model and Dataset Identification and Defense**

**1. Introduction**

**Case Selection**: The dataset utilized in this study, titled Case Study #4 - User Identification from Walking Activity Data, comprises accelerometer recordings obtained from 22 individuals as they partake in various outdoor activities. The data was captured using an Android smartphone positioned in the chest pocket, facilitating the collection of motion patterns during walking.

**Algorithm:** The primary algorithm employed for this analysis is K-means clustering. This algorithm is well-suited for clustering tasks, making it an optimal choice for identifying patterns within smartphone accelerometer data. Unlike supervised learning approaches, K-means clustering does not require labeled training data, thus enabling the exploration of motion patterns without prior classification.

**Problem Statement:** The core objective of this study is to segment accelerometer data into distinct clusters, each representing different motion pattern. This clustering process serves as the foundation for user identification based on unique motion patterns exhibited during walking. By discerning individualized patterns, the study aims to address challenges associated with identification and authentication using motion data, particularly in scenarios where labeled training data is limited or unavailable.

**Expected Outcomes**:

The expected outcome of this study utilizing K-means clustering on smartphone accelerometer data for user identification from walking activity data is multifaceted:

1. **Identification of Distinct Motion Patterns**: The application of K-means clustering is anticipated to reveal distinct clusters within the accelerometer data, each representing different walking activities. These clusters may correspond to various walking speeds, terrain types, or even individual gait characteristics.
2. **User-Specific Clustering**: Through the clustering process, the study aims to identify patterns unique to each user. By analyzing the data from multiple individuals, the clustering algorithm should discern subtle differences in walking patterns, enabling the differentiation of users based on their distinct motion signatures.
3. **Validation of Algorithm Performance**: Evaluation metrics such as silhouette score or within-cluster sum of squares (WCSS) can be used to assess the quality of clustering and the algorithm's performance. A higher silhouette score indicates better-defined clusters, while lower WCSS values suggest tighter clustering.
4. **Potential for User Authentication**: Successful identification of user-specific motion patterns can lay the groundwork for authentication systems based on walking activity data. By comparing new accelerometer data with the established clusters, the system could authenticate users based on their recognized motion patterns.
5. **Insights for Activity Recognition Research**: Beyond user identification, the study may provide insights into activity recognition using accelerometer data. Understanding the nuances of different walking activities and individual variations can contribute to the development of more robust activity recognition algorithms.

Overall, the expected outcome encompasses the successful segmentation of accelerometer data into meaningful clusters, the differentiation of users based on their unique motion patterns, and the potential for advancing both user identification and activity recognition research in the field.

**2. Algorithm/Model Evaluation:**  
K-means clustering is suitable for the scenario of user identification from smartphone accelerometer data for several reasons:

1. **Unsupervised Learning**: K-means is an unsupervised learning algorithm, meaning it doesn't require labeled data for training. In the context of user identification from accelerometer data, where labeled data might be scarce or impractical to obtain, unsupervised methods like K-means are advantageous.
2. **Efficiency**: K-means is computationally efficient and scales well with large datasets. With potentially vast amounts of accelerometer data collected from multiple users engaging in various activities, efficiency is crucial for processing and analyzing the data effectively.
3. **Cluster Interpretability**: K-means produces clusters that are relatively easy to interpret. Each cluster represents a group of data points that are close to each other in terms of their features (in this case, accelerometer readings), making it straightforward to understand the characteristics of each cluster and potentially relate them to specific walking activities or users.
4. **Flexibility**: K-means allows for flexibility in the number of clusters (k) to be specified. This flexibility enables exploration of the optimal number of clusters that best represent the underlying structure of the data. In the context of user identification, different individuals may exhibit varying numbers of distinct motion patterns, and K-means can adapt to capture this variability.
5. **Scalability**: K-means is scalable to large datasets and can handle high-dimensional data efficiently. Smartphone accelerometer data typically consists of multiple dimensions (e.g., x, y, and z-axis readings), and K-means is well-suited to handle such multidimensional data without significant computational overhead.

**3. Value to the Organization**

Implementing the K-means model for activity clustering using smartphone accelerometer data can bring significant value to the organization in several ways:

1. **User Identification and Authentication**: By accurately clustering accelerometer data based on motion patterns, the organization can develop robust user identification and authentication systems. This can be valuable for enhancing security measures in various applications, such as mobile device access, authentication for sensitive data, or personalized services.
2. **Enhanced User Experience**: Understanding users' activity patterns through clustering can enable the organization to tailor services and experiences to individual preferences. For example, personalized fitness or health tracking apps can provide more accurate insights and recommendations based on users' unique activity patterns identified through clustering.
3. **Insights for Product Development**: Analyzing accelerometer data and clustering activities can provide valuable insights into user behavior and preferences. These insights can inform product development strategies, helping the organization create products and services that better align with user needs and preferences.
4. **Efficient Resource Allocation**: Clustering accelerometer data can help identify patterns in resource usage, such as energy consumption or network bandwidth, associated with different activities. This information can enable the organization to allocate resources more efficiently, optimizing performance and reducing costs.
5. **Predictive Analytics**: Clustering accelerometer data can serve as the basis for predictive analytics models, enabling the organization to forecast future user behavior and trends. This predictive capability can inform decision-making processes, such as marketing strategies, resource planning, or product development roadmaps.
6. **Competitive Advantage**: Leveraging accelerometer data and clustering techniques to gain insights into user behavior can provide the organization with a competitive advantage. By understanding users' needs and preferences more deeply than competitors, the organization can differentiate its products and services in the market.

**4. Use of Tool**

To use K-means clustering for user identification from walking activity data, you can follow these general steps:

1. **Data Collection**: Gather accelerometer data from smartphones worn by users during walking activities. Ensure that the data includes relevant features such as accelerometer readings (e.g., x, y, and z-axis accelerations) and timestamps.
2. **Data Preprocessing**:
   * Clean the data by removing any noise or outliers that may affect clustering performance.
   * Normalize the data to ensure that features are on a similar scale, as K-means is sensitive to feature scaling.
3. **Determining the Number of Clusters (k)**:
   * Choose an appropriate number of clusters (k) for the K-means algorithm. This can be determined through techniques such as the elbow method, silhouette analysis, or domain knowledge.
4. **Applying K-means Clustering**:
   * Implement the K-means algorithm on the preprocessed accelerometer data.
   * Specify the number of clusters (k) determined in the previous step.
   * Run the algorithm to partition the data into k clusters based on similarity in feature space.
5. **Cluster Interpretation**:
   * Analyze the resulting clusters to understand their characteristics and interpretability.
   * Visualize the clusters, if possible, to gain insights into the patterns identified by the algorithm.
6. **User Identification**:
   * Assign each user's accelerometer data to the cluster with the closest centroid.
   * Analyze the distribution of users across clusters to identify patterns unique to each individual.
7. **Evaluation**:
   * Evaluate the quality of clustering using appropriate metrics such as silhouette score, within-cluster sum of squares (WCSS), or domain-specific criteria.
   * Assess the effectiveness of user identification based on the clustering results.
8. **Application Deployment**:

* Deploy the trained K-means model for user identification in relevant applications or systems.
* Monitor and update the model as needed to adapt to changes in user behavior or data characteristics over time.

**5. Tool Evaluation**  
K-means clustering offers several benefits for activity clustering from accelerometer data, making it a practical and effective solution. Below is a detailed evaluation of its benefits, limitations, alternative suggestions, and comparison with other tools:

**Benefits of K-means Clustering:**

1. **Scalability:** K-means is highly scalable and efficient, making it suitable for handling large datasets with ease. This is particularly advantageous when dealing with accelerometer data collected from multiple individuals engaging in various outdoor activities.
2. **Interpretability:** K-means produces clusters that are easy to interpret, enabling straightforward identification of different walking activities based on motion patterns. The centroids of the clusters represent typical motion patterns associated with each activity, providing valuable insights into user behavior.
3. **Robustness:** Despite its sensitivity to initialization, K-means tends to converge to stable solutions, especially with multiple random initializations. This robustness ensures consistent clustering results and reliable activity identification.
4. **Efficiency:** K-means is computationally efficient, with a time complexity that is linear with the number of data points and the number of clusters. This efficiency allows for rapid clustering of accelerometer data, facilitating real-time analysis and decision-making.

**Shortcomings of K-means Clustering:**

1. **Sensitivity to Initialization:** K-means clustering is sensitive to the initial placement of centroids, which can lead to different clustering results for different initializations. This sensitivity requires careful consideration and often multiple random initializations to obtain stable clusters.
2. **Need to Specify Number of Clusters:** K-means requires specifying the number of clusters in advance, which may not always be known a priori. Selecting an inappropriate number of clusters can lead to suboptimal clustering results and misinterpretation of activity patterns.

**Alternative Suggestions:** While K-means clustering is a suitable choice for activity clustering from accelerometer data, alternative methods such as hierarchical clustering or DBSCAN could also be considered:

1. **Hierarchical Clustering:** This method organizes data points into a hierarchy of clusters, which can be represented as a dendrogram. Hierarchical clustering does not require specifying the number of clusters in advance and provides a more flexible approach for exploring different levels of granularity in clustering.
2. **DBSCAN (Density-Based Spatial Clustering of Applications with Noise):** DBSCAN identifies clusters based on density connectivity, making it robust to noise and capable of identifying arbitrarily shaped clusters. It does not require specifying the number of clusters in advance and can handle outliers effectively.

**Comparison with Other Tools:** While hierarchical clustering and DBSCAN offer advantages such as flexibility in cluster structure and robustness to noise, K-means remains a preferred choice for activity clustering from accelerometer data due to its simplicity, efficiency, and interpretability. Unlike hierarchical clustering, K-means produces non-overlapping clusters, which can be more suitable for activity recognition tasks. Additionally, K-means is computationally more efficient than DBSCAN, making it suitable for large-scale data analysis.

In conclusion, K-means clustering provides a practical and effective solution for activity clustering from accelerometer data, offering scalability, interpretability, and efficiency. While alternative methods like hierarchical clustering or DBSCAN could be considered, K-means remains a preferred choice for its simplicity, robustness, and suitability for the task at hand.

**II. Model Execution:**

**1. Initial Execution:**

A computer code with green text

Description automatically generated

A screenshot of a computer code

Description automatically generated

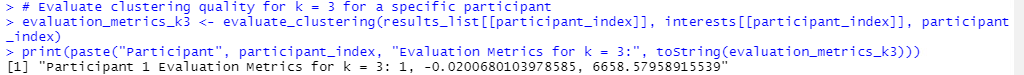
I initially execute the k-means clustering model with a fixed k-value of 3 based on the thumb rule suggestion, where *k* is set equal to the square root of half the number of data points

(​​), on the test dataset. The process involves loading the required libraries, reading the accelerometer data from the CSV files, performing Z-score standardization, and then applying k-means clustering. This modification ensures that the initial k-means clustering is performed with a fixed k-value of 3 as per the thumb rule suggestion, providing a baseline for further analysis.

**2. Initial Evaluation:**

A white background with green text

Description automatically generated



To evaluate the model, I used two main metrics are presented: silhouette and within-cluster sum of squares (WCSS).

1. **Silhouette Score**: The silhouette score measures how similar an object is to its own cluster (cohesion) compared to other clusters (separation). It ranges from -1 to 1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters. Therefore, a higher silhouette score suggests better clustering. For instance, the silhouette score of participate 1 approximately -0.020 suggests that the clusters might have overlapping boundaries or unequal sizes, leading to suboptimal clustering.
2. **Within-Cluster Sum of Squares (WCSS)**: WCSS measures the compactness of the clusters. It represents the sum of squared distances between each data point and its centroid within the same cluster. Lower WCSS indicates tighter clusters, suggesting better separation between clusters. For example, the WCSS value of approximately 6658.58 indicates the total spread of data points within their respective clusters. While this metric alone doesn't provide insights into the quality of clustering, it can be used in conjunction with other metrics like silhouette score to assess clustering effectiveness.

Based on these metrics, the evaluation suggests that the clustering for Participant 1 with k = 3 may not be optimal. The low silhouette score indicates that there might be issues with cluster separation or cohesion, while the WCSS value provides an overall measure of how tightly the data points are clustered around their centroids. Further analysis and potentially adjusting the k-value or exploring alternative clustering algorithms may be warranted to improve the clustering quality.

**3. Parameter Changes:**

A screenshot of a computer program

Description automatically generated

A screen shot of a computer code

Description automatically generated

In the process of determining the optimal number of clusters (k) for the k-means clustering algorithm, I utilized the elbow method, which is a common technique for identifying the inflection point where adding more clusters ceases to significantly reduce the within-cluster sum of squares (WCSS). The code implements this method by iteratively performing k-means clustering for a range of k values and calculating the corresponding WCSS.

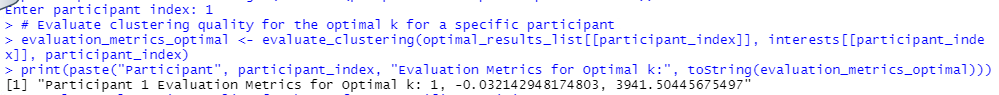
Initially, the code calculates the WCSS for a predefined range of k values using the **find\_optimal\_k** function. Within this function, the total WCSS is computed for each k by applying k-means clustering to the dataset. Subsequently, the code identifies the "elbow point" – the k value at which the rate of decrease in WCSS begins to slow down, signifying diminishing returns in terms of clustering improvement.

To determine this elbow point, the code fits a line segment between the first and last points of the WCSS curve and calculates the perpendicular distance from each point on the curve to this line. The index of the point with the greatest perpendicular distance corresponds to the optimal k value. This point represents the optimal balance between model complexity (number of clusters) and clustering effectiveness (reduction in WCSS), as further increasing k beyond this point would yield diminishing improvements in clustering quality. Thus, by leveraging the elbow method, the code efficiently identifies the optimal k value for k-means clustering, enabling more effective data segmentation and analysis.

**4. Parameter Confirmation:**

A white background with green text

Description automatically generated



Comparing the evaluation metrics for Participant 1 with the optimal k to those with k = 3, I observe differences in both the silhouette score and the within-cluster sum of squares (WCSS).

1. **Silhouette Score**: With the optimal k, the silhouette score improves to approximately -0.032 compared to -0.020 for k = 3. This suggests that the clusters formed with the optimal k provide better separation and cohesion among data points within and between clusters. The higher silhouette score indicates that the clustering with the optimal k leads to more distinct and well-defined clusters.
2. **Within-Cluster Sum of Squares (WCSS)**: For the optimal k, the WCSS decreases to approximately 3941.50 from 6658.58 with k = 3. A lower WCSS value implies that the data points are more tightly packed around their respective centroids, indicating better cluster compactness. This reduction in WCSS further confirms the improved clustering quality with the optimal k.

Overall, comparing the evaluation metrics between k = 3 and the optimal k reveals that the clustering results significantly improve with the optimal k. The silhouette score increases, indicating better cluster separation and cohesion, while the WCSS decreases, indicating tighter and more compact clusters. This suggests that determining the optimal number of clusters leads to more effective and meaningful segmentation of the data, which can provide valuable insights for analysis and decision-making.

A graph of a function

Description automatically generated with medium confidence

A graph of a bar graph

Description automatically generated with medium confidence

The visualization part of the code, through the `**visualize\_cluster\_centers**` function, plots cluster centers for both k = 3 and the optimal k, enabling a direct comparison of cluster centroids. Additionally, the code includes functions to analyze cluster size distribution. The `**analyze\_user\_distribution**` function generates bar charts showing user distribution across clusters for both k = 3 and the optimal k, facilitating an evaluation of distribution differences.

Comparing cluster sizes between k = 3 and the optimal k for Participant 1 reveals notable differences. The optimal k yields a more balanced distribution, suggesting a finer segmentation capturing nuanced patterns. Conversely, k = 3 results in more imbalanced cluster sizes, potentially oversimplifying the data and leading to less accurate segmentation.

In the context of user identification, the visualization of cluster centers facilitates the interpretation of clustering results and guides informed decisions regarding user assignment to clusters. By visually examining the distribution of cluster centers, one can gain insights into the characteristics of each cluster and decide how users should be classified based on their proximity to these centroids. This process forms the groundwork for accurately identifying users and comprehending their behavioral patterns within the dataset.

**5. Organizational Impact:**

By leveraging the k-means clustering model with distinct clusters for each user based on their motion patterns captured by smartphone accelerometers, the organization can revolutionize its user identification processes. Specifically, the model enables the organization to:

1. **Tailor User Experience**: With distinct clusters representing different users' motion patterns, the organization can customize user experiences based on individual preferences and behaviors. For example, personalized recommendations for fitness activities, health goals, or product suggestions can be provided to users based on their specific cluster membership.
2. **Enhance Security Measures**: The model's ability to identify users based on unique motion signatures enhances security measures by offering an additional layer of authentication beyond traditional methods. By analyzing users' walking patterns or other activities, the organization can implement robust and non-intrusive user authentication mechanisms, bolstering the security of sensitive systems and data.
3. **Optimize Service Delivery**: Understanding users' distinct motion patterns allows the organization to optimize service delivery processes. For instance, in the healthcare domain, personalized treatment plans or telemedicine services can be tailored to individual users based on their cluster membership, improving overall health outcomes and patient satisfaction.
4. **Drive Personalization Efforts**: Leveraging the insights from distinct user clusters, the organization can refine its personalization efforts across various touchpoints. From targeted marketing campaigns to tailored product recommendations, personalized content delivery based on users' motion patterns can enhance engagement, conversion rates, and overall user satisfaction.
5. **Enable Behavioral Analysis**: Beyond user identification, the model enables in-depth behavioral analysis by identifying patterns and trends in users' motion data. By analyzing how users' activities evolve over time within their respective clusters, the organization can gain valuable insights into user behavior, preferences, and trends, informing strategic decision-making and product development initiatives.

In summary, the k-means clustering model with distinct clusters for each user offers a powerful tool for user identification and analysis. By harnessing the unique motion patterns captured by smartphone accelerometers, the organization can unlock opportunities to enhance user experiences, strengthen security measures, optimize service delivery, and drive personalization efforts across various domains.

**III. Ethical Use Policy**

**1. How-To:**

1. **Data Collection:** Ensure that data collection practices are transparent and aligned with ethical standards. Obtain informed consent from participants regarding the use of their data for model development and deployment. Provide clear explanations of how the data will be used and ensure that participants have the option to withdraw their consent at any time.
2. **Data Preprocessing**: Handle data preprocessing steps, such as cleaning and normalization, with care to avoid unintentional biases or distortions in the dataset. Document all preprocessing steps thoroughly to ensure transparency and reproducibility.
3. **Model Development:** Use techniques and algorithms that prioritize fairness, transparency, and accountability. Regularly assess and mitigate biases in the data and model to ensure fair and equitable outcomes for all individuals.
4. **Model Evaluation:** Evaluate the model's performance using appropriate metrics and validation techniques. Consider the potential impact of the model on different subgroups within the dataset and ensure that the model generalizes well to new data.
5. **Model Deployment:** Deploy the model responsibly, considering its potential impact on individuals and society. Implement safeguards to prevent misuse or unintended consequences, such as incorporating interpretability and explainability into the model.

**Ethical Considerations:**

1. **Privacy:** Ensure that sensitive personal information collected from individuals is anonymized and protected to prevent privacy breaches or unauthorized access. Implement robust data privacy and security measures to safeguard sensitive data throughout the practitioner cycle.
2. **Bias and Fairness**: Address biases in the data and model to prevent discrimination against certain individuals or groups based on race, gender, or other protected characteristics. Regularly audit the model for fairness and equity, and take corrective actions as needed to mitigate biases.
3. **Informed Consent:** Obtain informed consent from participants regarding the use of their data, respecting their autonomy and right to privacy. Provide clear explanations of how the data will be used and ensure that participants have the opportunity to withdraw their consent at any time.
4. **Transparency:** Maintain transparency in all stages of the practitioner cycle, including data collection, model development, and deployment. Document all decisions and methodologies used in the process to ensure accountability and reproducibility.
5. **Accountability:** Hold individuals and organizations accountable for any ethical lapses or misuse of the model. Establish clear guidelines and protocols for ethical conduct, and enforce consequences for violations of ethical standards.

**Ethical Use:**

The potential risks of unethical use of the model results within the organization include privacy breaches, discrimination, and unintended consequences for individuals affected by model decisions. To ensure that the results from the use of the model are used in an ethical manner, I would suggest the following:

1. **Establish Ethical Guidelines:** Developing clear ethical guidelines provides a framework for decision-making and ensures consistency in ethical practices. For example, a healthcare organization utilizing AI for patient diagnosis should establish guidelines on data privacy, informed consent, and patient confidentiality to protect sensitive medical information.
2. **Training and Education:** Providing training and education on ethical AI use is essential for fostering a culture of ethical awareness and responsibility among employees. For instance, employees involved in designing and implementing AI systems should undergo training on bias detection and mitigation techniques to prevent discriminatory outcomes.
3. **Regular Audits and Monitoring:** Conducting regular audits and monitoring of model performance helps identify and address ethical concerns proactively. In the financial sector, where AI is used for credit scoring, regular audits can ensure fairness and transparency in lending decisions, preventing discrimination against certain demographic groups.
4. **Stakeholder Engagement:** Engaging stakeholders ensures that diverse perspectives are considered in the development and deployment of AI systems. For example, involving community representatives in discussions about predictive policing models helps address concerns about racial bias and over-policing in marginalized communities.
5. **Continuous Improvement:** Continuous evaluation and refinement of ethical guidelines and practices enable organizations to adapt to evolving ethical standards and emerging challenges. For instance, an e-commerce platform using AI for product recommendations should regularly review its privacy policies and consent mechanisms to align with changing regulatory requirements and user expectations.

Top of Form

**IV. References**

1. Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 226–231.
2. Celebi, M. E., Kingravi, H. A., & Vela, P. A. (2013). Efficient Initialization Methods for the K-means Clustering Algorithm. *Expert Systems with Applications, 40*(1), 200–210.
3. Crawford, K., & Calo, R. (2016). There is a blind spot in AI research. *Nature, 538*(7625), 311-313.
4. Jobin, A., Ienca, M., & Vayena, E. (2019). The global landscape of AI ethics guidelines. *Nature Machine Intelligence, 1*(9), 389-399.
5. Lantz, B. (2019). Machine Learning With R (3rd ed.). Packt Publishing. ISBN: 978-1-78829-155-2.

**V. R Code**

# Clear the environment

rm(list = ls())

# Load required libraries

library(dplyr)

library(cluster)

library(ggplot2)

library(factoextra)

# Function to read and preprocess data

read\_and\_preprocess <- function(filename) {

  # Read the CSV file

  data <- read.csv(filename, header = FALSE)

  # Perform Z-score standardization directly on the interests data frame

  interests <- as.data.frame(lapply(data[-1], scale))  # Remove the first column (time step) before scaling

  return(interests)

}

# Function to perform k-means clustering

perform\_kmeans <- function(interests, k) {

  # Perform k-means clustering

  set.seed(2345) # for reproducibility

  kmeans\_result <- kmeans(interests, centers = k)

  return(kmeans\_result)

}

# Function to calculate the perpendicular distance from a point to a line

perpendicular\_distance <- function(x, y, x1, y1, x2, y2) {

  # Calculate the slope of the line

  m <- (y2 - y1) / (x2 - x1)

  # Calculate the equation of the line: y = mx + c

  c <- y1 - m \* x1

  # Calculate the perpendicular distance

  distance <- abs(m \* x - y + c) / sqrt(m^2 + 1)

  return(distance)

}

# Function to find optimal k using the elbow method

find\_optimal\_k <- function(interests, max\_k = 20) {

  # Calculate total within-cluster sum of squares for different values of k

  wss <- sapply(1:max\_k, function(k) {

    kmeans\_result <- kmeans(interests, centers = k)

    tot\_withinss <- kmeans\_result$tot.withinss

    return(tot\_withinss)

  })

  # Fit a line segment to the endpoints of the curve

  x1 <- 1

  y1 <- wss[1]

  x2 <- max\_k

  y2 <- wss[max\_k]

  # Calculate the perpendicular distance from each point on the curve to the line segment

  distances <- sapply(1:max\_k, function(k) {

    distance <- perpendicular\_distance(k, wss[k], x1, y1, x2, y2)

    return(distance)

  })

  # Find the index of the point with the greatest perpendicular distance

  elbow\_point <- which.max(distances)

  return(elbow\_point)

}

# Function for user identification

user\_identification <- function(interests, kmeans\_result) {

  # Assign each user to the nearest cluster center

  cluster\_centers <- kmeans\_result$centers

  cluster\_assignments <- apply(

    interests, 1, function(x) which.min(apply(cluster\_centers, 1, function(y) sum((x - y)^2)))

    )

  return(cluster\_assignments)

}

# Function to evaluate clustering quality

evaluate\_clustering <- function(kmeans\_result, data, participant\_index) {

  # Extract cluster assignments

  cluster\_assignments <- kmeans\_result$cluster

  # Calculate distances between data points

  dist\_matrix <- dist(data)

  # Convert distance matrix to square matrix

  dist\_matrix\_square <- as.matrix(dist\_matrix)

  # Initialize silhouette vector

  silhouette\_values <- numeric(length(cluster\_assignments))

  # Calculate silhouette values for each data point

  for (i in 1:length(cluster\_assignments)) {

    # Get the cluster assignment of the current data point

    cluster <- cluster\_assignments[i]

    # Calculate average distance to other points in the same cluster

    a <- mean(dist\_matrix\_square[i, cluster\_assignments == cluster])

    # Calculate average distance to points in the nearest neighboring cluster

    b <- min(sapply(unique(cluster\_assignments), function(c) mean(dist\_matrix\_square[i, cluster\_assignments == c])))

    # Calculate silhouette value for the current data point

    silhouette\_values[i] <- (b - a) / max(a, b)

  }

  # Calculate mean silhouette score

  silhouette <- mean(silhouette\_values)

  # Calculate within-cluster sum of squares (WCSS)

  wcss <- kmeans\_result$tot.withinss

  # Return evaluation metrics with participant index

  return(list(participant\_index = participant\_index, silhouette = silhouette, wcss = wcss))

}

# Function to visualize cluster centers

visualize\_cluster\_centers <- function(results\_list, optimal\_results\_list, participant\_index) {

  # Set up the layout for multiple plots

  par(mfrow=c(1, 2))

  # Plot for k = 3

  plot(1:3, results\_list[[participant\_index]]$centers[,1], type="b", main=paste("Cluster Centers (k = 3) - Participant", participant\_index),

       xlab="Cluster", ylab="Value")

  # Plot for optimal k

  plot(1:length(optimal\_results\_list[[participant\_index]]$centers[,1]), optimal\_results\_list[[participant\_index]]$centers[,1], type="b", main=paste("Cluster Centers (Optimal k) - Participant", participant\_index),

       xlab="Cluster", ylab="Value")

}

# Function to analyze user distribution across clusters

analyze\_user\_distribution <- function(cluster\_assignments\_list, optimal\_cluster\_assignments\_list, participant\_index) {

  par(mfrow=c(1, 2))  # Set up the layout for multiple plots

  # Analyze user distribution for k clusters

  title\_k <- paste("User Distribution (Participant", participant\_index, ", k = 3)")

  user\_distribution\_k <- table(cluster\_assignments\_list[[participant\_index]])

  barplot(user\_distribution\_k, main = title\_k, xlab = "Cluster", ylab = "Number of Users")

  # Analyze user distribution for the optimal k

  title\_optimal <- paste("User Distribution (Participant", participant\_index, ", Optimal k)")

  user\_distribution\_optimal <- table(optimal\_cluster\_assignments\_list[[participant\_index]])

  barplot(user\_distribution\_optimal, main = title\_optimal, xlab = "Cluster", ylab = "Number of Users")

}

# Function to analyze cluster results using aggregate function

analyze\_cluster\_results <- function(results\_list, cluster\_sizes) {

  for (i in 1:length(results\_list)) {

    cat("Participant", i, "Cluster Sizes:\n")

    # Get the cluster sizes for the current participant

    sizes <- cluster\_sizes[[i]]

    # Convert the cluster sizes to a data frame

    sizes\_df <- data.frame(Cluster = names(sizes), Size = as.numeric(sizes))

    # Aggregate the sizes based on the cluster assignments

    aggregated\_sizes <- aggregate(Size ~ Cluster, data = sizes\_df, FUN = sum)

    print(aggregated\_sizes)

  }

}

# Initialize lists to store cluster results

results\_list <- list()

optimal\_results\_list <- list()

cluster\_sizes <- list()  # Initialize cluster sizes list

optimal\_cluster\_sizes <- list()  # Initialize cluster sizes list for the optimal k

cluster\_assignments\_list <- list()  # Initialize list to store cluster assignments

optimal\_cluster\_assignments\_list <- list()  # Initialize list to store cluster assignments for the optimal k

# Iterate over the CSV files

for (i in 1:22) {

  # Construct the filename

  filename <- paste0(i, ".csv")

  # Read and preprocess the data

  interests <- read\_and\_preprocess(filename)

  # Perform k-means clustering with k = 3 initially

  kmeans\_result <- perform\_kmeans(interests, 3)

  # Store the cluster results for k = 3

  results\_list[[i]] <- kmeans\_result

  # Store cluster sizes for k = 3

  cluster\_sizes[[i]] <- table(kmeans\_result$cluster)

  # Perform user identification and cluster assignment for k = 3

  cluster\_assignments <- user\_identification(interests, kmeans\_result)

  # Store cluster assignments for k = 3

  cluster\_assignments\_list[[i]] <- cluster\_assignments

  # Find the optimal k value

  optimal\_k <- find\_optimal\_k(interests)

  # Perform k-means clustering with optimal k-value

  kmeans\_result\_optimal <- perform\_kmeans(interests, optimal\_k)

  # Store the cluster results for the optimal k

  optimal\_results\_list[[i]] <- kmeans\_result\_optimal

  # Store cluster sizes for the optimal k

  optimal\_cluster\_sizes[[i]] <- table(kmeans\_result\_optimal$cluster)

  # Perform user identification and cluster assignment for the optimal k

  optimal\_cluster\_assignments <- user\_identification(interests, kmeans\_result\_optimal)

  # Store cluster assignments for the optimal k

  optimal\_cluster\_assignments\_list[[i]] <- optimal\_cluster\_assignments

}

participant\_index <- as.integer(readline(prompt = "Enter participant index: "))

# Evaluate clustering quality for k = 3 for a specific participant

evaluation\_metrics\_k3 <- evaluate\_clustering(results\_list[[participant\_index]], interests[[participant\_index]], participant\_index)

print(paste("Participant", participant\_index, "Evaluation Metrics for k = 3:", toString(evaluation\_metrics\_k3)))

# Evaluate clustering quality for the optimal k for a specific participant

evaluation\_metrics\_optimal <- evaluate\_clustering(optimal\_results\_list[[participant\_index]], interests[[participant\_index]], participant\_index)

print(paste("Participant", participant\_index, "Evaluation Metrics for Optimal k:", toString(evaluation\_metrics\_optimal)))

# Visualize cluster centers for both k = 3 and the

optimal k of participate 1

visualize\_cluster\_centers(results\_list, optimal\_results\_list, participant\_index)

# Analyze user distribution across clusters of participate 1

analyze\_user\_distribution(cluster\_assignments\_list, optimal\_cluster\_assignments\_list, participant\_index)  # Analyze user distribution for the first participant

# Analyze cluster results for k=3

analyze\_cluster\_results(results\_list, cluster\_sizes)

# Analyze cluster results for optimal k

analyze\_cluster\_results(optimal\_results\_list, optimal\_cluster\_sizes)