

A Hybrid Movie Recommendation System using Particle Swarm Optimization and K-means Clustering Algorithm

By: Daniel Aschalew Demissie



A Proposal Submitted to the Department of Computer Science and Engineering, School of Electrical Engineering and Computing.

Office of Graduate Studies
Adama Science and Technology University (ASTU)

Adama, Ethiopia,

June 22, 2020

A Hybrid Movie Recommendation System using Particle Swarm Optimization and K-means Clustering Algorithm

By: Daniel Aschalew Demissie

Advisor: Teklu Uregessa (PhD)

A Proposal Submitted to the Department of Computer Science and Engineering, School of Electrical Engineering and Computing.

Office of Graduate Studies

Adama Science and Technology University (ASTU)

Adama, Ethiopia

June 22, 2020

Declaration

I hereby declare that this MSc thesis is my original work and has not been presented as a partial degree requirement for a degree in any other university, and that all sources of materials used for the thesis have been fully acknowledged.

Name: Daniel Aschalew

Signature: _____

This thesis has been submitted for examination with my approval as thesis advisor.

Name: Teklu Uregessa (PhD)

Signature: _____

Date of Submission: 6.22.2020

Approval of Board of Examiners

We, the undersigned, members of the Board of Examiners of the final open defense by Daniel Aschalew have read and evaluated his/her thesis entitled “A Hybrid Movie Recommendation System using Particle Swam Optimization and K-means Clustering Algorithm” and examined the candidate. This is, therefore, to certify that the thesis has been accepted in partial fulfillment of the requirement of the Degree of Masters of Science in Computer Science and Engineering

_____ Supervisor/Advisor	_____ Signature	_____ Date
_____ Chairperson	_____ Signature	_____ Date
_____ Internal Examiner	_____ Signature	_____ Date
_____ External Examiner	_____ Signature	_____ Date

Acknowledgement

Foremost, I would like to thank the Ultimate God for his almighty, and his mother Saint Merry. I am thankful Lord for everything that you allow me to pass through all happiness, sadness, success and failure. I am also thankful for the strength you gave me to make bold and scary decisions and to learn from the lessons that come from those decisions.

I respect and thank Teklu Uregessa (PhD.), for providing me an opportunity to do the research work in Recommendation Systems and giving me all support and guidance, which made me complete the research duty. I am extremely thankful to him for providing such nice support and guidance, although he had a busy schedule managing the coordinating affairs.

I would also like to acknowledge the members of Smart Software SIG of the CSE department. Including the SIG supervisor Mesifn Abebe (PhD.), for his continued commitment to guide and support this research from its beginning to the end. And to all members who sat on each phase presentation panels for their inputs which greatly shaped and improved the work. I would like to thanks all my classmates who are working on their master's thesis for their timely help, idea, and support until the completion of my thesis.

My sincere thanks also go to all my family: Aschalew, Etaferaw, Anteneh, Seble, Genet, Mesfin, if you were not loving and supporting, I would not have been able to do this work very easily. Dear Father, God give you all the gifts you gave me and thank you for all the wisdom, bravery, persistence and strength you showed me and for all that you have done for me while you are alive.

This thesis is dedicated to my father, Aschalew Demissie.

I would like to thank all ASTU's CSE department staff members, whose helps me by motivating and helping me to deliver the required result.

Finally, it is my radiant sentiment to place on record my best regards, deepest sense of gratitude to my friends in CSE Department for their timely help and wholehearted support rendered to me in the completion of my thesis.

Table of Contents

Contents

Declaration.....	i
Approval of Board of Examiners.....	ii
Acknowledgement	iii
List of Tables	ix
List of Figures	x
List of Equations.....	xii
List of Abbreviation.....	xiii
Abstract	xiv
Chapter one	1
1. Introduction	1
1.1. Overview	1
1.2. Motivation	2
1.3. Statement of the Problem	3
1.4. Objectives.....	4
1.4.1. General Objective	4
1.4.2. Specific Objective	4
1.5. Data Collection and Analysis.....	4
1.6. Methodology of the Study.....	4
1.7. Scope and Limitations.....	5
1.8. Application of Results.....	6
1.9. Thesis Organization.....	7
Chapter Two.....	8
2. Literature Review	8

2.1.	Overview	8
2.2.	Recommendation System.....	8
2.3.	Phases of Recommendation system	9
2.3.1.	Information Collection Phase	10
2.3.2.	Learning Phase.....	11
2.3.3.	Prediction or Recommendation Phase	11
2.4.	Methods of recommendation system.....	11
2.4.1.	Content-based Filtering.....	12
2.4.2.	Collaborative Filtering.....	13
2.4.3.	Hybrid Technique	16
2.4.4.	Knowledge-based Technique.....	17
2.5.	Data Pre-processing.....	17
2.5.1.	Data Conversion Mechanisms	18
2.6.	Machine learning.....	19
2.6.1.	Supervised Machine Learning	19
2.6.2.	Unsupervised Machine Learning	19
2.6.3.	Semi-supervised Machine Learning	20
2.7.	Clustering in Recommendation System	20
2.7.1.	Partitional Method	21
2.7.2.	Hierarchical Method	23
2.7.3.	Density-based Method	23
2.7.4.	Grid-based Method	23
2.8.	Optimization Algorithms.....	24
2.9.	Movie Recommendation System	26
2.10.	Review Papers	26

2.11. Summary	28
Chapter Three.....	30
3. Research Methodology	30
3.1. Data Collection Mechanism	30
3.1.1. Search Strategy	31
3.1.2. Inclusion and Exclusion Criteria.....	31
3.1.3. Dataset Used	33
3.2. Review Papers	34
3.3. Proposed solution or Framework	35
3.4. Evaluation Matrices of the Model.....	36
Chapter Four	39
4. Proposed Hybrid Recommendation System.....	39
4.1. Proposed Hybrid Model	39
4.1.1. Pre-processing.....	40
4.1.2. K-means Clustering Algorithm.....	43
4.1.3. Particle Swarm Optimization Algorithm (PSO)	47
4.1.4. Hybrid Particle Swarm Optimization Algorithm and K-means.....	51
4.1.5. Recommendation Algorithm.....	53
Chapter Five.....	55
5. Implementation of the Proposed Model	55
5.1. Implementation Environment.....	55
5.2. Deployment Environment	57
5.3. Dataset Description	57
5.4. Pre-processing Implementation.....	58
5.4.1. Feature Selection Implementation	58

5.4.2.	Data Integration Implementation	59
5.4.3.	Data Cleaning Implementation	59
5.4.4.	Data Conversion Implementation	59
5.5.	K-means Clustering Algorithm Implementation.....	60
5.5.1.	Determining K (number of Clusters)	60
5.5.2.	K-means Clustering Algorithm.....	62
5.6.	Hybrid Particle Swarm Optimization and K-means Clustering Algorithm (PSO-KM) Implementation.....	62
5.7.	Model Evaluation Implementation.....	64
5.8.	Popularity-based Recommendation Algorithm Implementation.....	65
Chapter Six.....		68
6.	Result and Discussion.....	68
6.1.	Data Pre-processing Result	68
6.1.1.	Feature Selection Stage Result.....	68
6.1.2.	Data Integration Stage Result	69
6.1.3.	Data Cleaning Stage Result	69
6.1.4.	Data Conversion Stage Result	69
6.2.	Model (K-means and Hybrid PSO-KM) Evaluation Result	71
6.2.1.	K-means Clustering Algorithm Result.....	71
6.2.2.	Hybrid PSO-KM (Particle Swarm Optimization and K-means Clustering) Algorithm Result	75
6.2.3.	Popularity-based Recommendation Algorithm Result	77
6.3.	Discussion	80
Chapter Seven		84
7.	Conclusion, Recommendation and Future Work	84
7.1.	Conclusion.....	84

7.2.	Recommendation.....	85
7.3.	Future Work	85
	References	86
	Appendixes	93
1.	Appendix 1: Hybrid PSO-KM Algorithm and K-means Algorithm	93
2.	Appendix 2: Particle Search Algorithm	97
3.	Appendix 3: Particle Swarm Optimization Algorithm (PSO).....	99

List of Tables

Table 5.1 Description the Tools and Python Packages Used During the Implementation	56
Table 6.1 Comparisons of K-means Algorithm and Hybrid PSO-KM Algorithm using 7 clusters	80
Table 6.2 Comparisons of K-means Algorithm and Hybrid PSO-KM Algorithm using 8 clusters	81
Table 6.3 The comparisons of results of the two algorithms using 7 and 8 clusters	82

List of Figures

Figure 1.1 Work flow of the research	5
Figure 2.1 Recommendation phases	10
Figure 2.2 . Recommendation Techniques	12
Figure 2.3 Collaborative filtering	13
Figure 2.4 Cluster Formation.....	21
Figure 3.1 Literature review path.....	32
Figure 4.1 Proposed Hybrid Architecture	40
Figure 4.2 Pre-processing steps	41
Figure 4.3 k-means clustering algorithm basic process	43
Figure 4.4 Flow chart of k-means clustering algorithm basic process.	47
Figure 4.5 Flow chart of PSO algorithm basic process	51
Figure 4.6 Hybrid PSO-K-means algorithm steps	52
Figure 4.7 Recommendation Algorithm Steps.....	54
Figure 5.1 Python Code for Loading the Dataset	58
Figure 5.2 Python Code for Deleting unwanted Columns.....	58
Figure 5.3 Python Code for Deleting unwanted Columns.....	59
Figure 5.4 Python Code for Deleting Rows with Missing Values.....	59
Figure 5.5 Python Code for Converting Genre values to Columns	60
Figure 5.6 Python Code for Initializing Genre values	60
Figure 5.7 Python Code for Deleting Genres and Title Column	60
Figure 5.8 Python Code for Determining K by using Elbow Method.....	61
Figure 5.9 Python Code for visualizing Elbow points against number of clusters.....	61
Figure 5.10 Python Code for Creating Clusters by using K-means Algorithm.....	62
Figure 5.11 Python Code for Importing the required header files.....	63
Figure 5.12 Python Code for Implementing PSO-KM algorithm.....	64
Figure 5.13 Python Code for Implementing Silhouette Score.	65
Figure 5.14 Python Code for Implementing Sum-of Squares-Error (SSE)	65
Figure 5.15 Python Code for Implementing Quantization Error	65
Figure 5.16 Python Code for Implementing merging data1 with data dataset	66
Figure 5.17 Python Code for Implementing merging data2 with recommend1 dataset	66

Figure 5.18 Python Code for Implementing for filtering and ranking movie lists	67
Figure 6.1 data1 dataset after Feature Selection Stage	68
Figure 6.2 data2 dataset after Feature Selection Stage	68
Figure 6.3 data dataset after Data Integration Stage	69
Figure 6.4 data dataset after Data Conversion Stage	70
Figure 6.5 data dataset after Data Conversion Stage	70
Figure 6.6 data dataset after Data Pre-processing Stage.....	70
Figure 6.7 The Result of Elbow Method	72
Figure 6.8 The Result of Elbow Method	73
Figure 6.9 The Result of K-means Clustering	74
Figure 6.10 The Result of K-means Clustering Algorithm Evaluation	75
Figure 6.11 The Result of Hybrid PSO-KM Algorithm Evaluation.....	76
Figure 6.12 The Result Hybrid PSO-KM Algorithm Evaluation	77
Figure 6.13 The Result data dataset after average movie rating and Title are included.....	77
Figure 6.14 The Result data dataset after all the movies the user didn't rate are selected	78
Figure 6.15 The Top 20 Movies Recommended by Popularity-based Recommendation Algorithm	79

List of Equations

Equation 3.1 The computation of Silhouette Score	37
Equation 3.2 The computation of SSE Criterion	38
Equation 3.3 The computation of quantization error	38
Equation 4.1 The Computation of objective function.....	44
Equation 4.2 The Computation of Euclidean Distance.....	45
Equation 4.3 The Computation of new centroid	45
Equation 4.4 Equation 4.4. the Computation of Euclidean Distance.....	46
Equation 4.5 The Computation of new centroid	46
Equation 4.6 The Computation of particle's best position	48
Equation 4.7 The Computation of particle's personal best position	49
Equation 4.8 The Computation of the fitness of particles	49

List of Abbreviation

KM	K-means Algorithm
PSO	Particle Swarm Optimization
PSO-KM	Hybrid Particle Swarm Optimization and K-means Algorithm
ML	Machine Learning
AI	Artificial Intelligence
SSE	Sum of Squared Error

Abstract

On the Internet, where the number of choices is overwhelming, priorities need to be prioritized and relevant information provided efficiently to make selection easier and to minimize the problem of information overload, which has created a potential problem for many Internet users. Recommender systems solve this problem by searching through large amounts of dynamically generated information to provide users with personalized content and services. This essentially makes recommendation systems a central component of websites and e-commerce applications. Nowadays, cold-start and scalability problems are the researchable areas of recommendation systems. Solving these two problems will result in personalized recommendation and increased recommendation accuracy. To solve these problems, this research paper focuses on movie recommendation systems, the main goal of which is to propose a recommendation system through data clustering and computational intelligence. In this research article a novel recommendation system was discussed, which is a Hybrid Particle Swarm Optimization (PSO) and k-means algorithm (PSO-KM) applied to the MovieLens (ml-latest-small) dataset. This dataset first passed through the successive stages of Pre-processing process and produced quality and numerical data. After the quality and numerical data is produced, the Elbow Method determined the number of clusters or K as initial seeds of K-means clustering algorithm. Then, the Hybrid PSO-KM algorithm runs on the dataset and produced compact and homogenous clusters. [The result of evaluation matrices show that the proposed Hybrid algorithm produced 0.93% more compact clusters and 0.97% better separation between neighboring clusters than the k-means clustering algorithm.](#)

Key words: Elbow Method, Clustering, K-means, Optimization, Particle Swarm Optimization and Hybrid Particle Swarm Optimization and K-means

Chapter one

1. Introduction

1.1. Overview

In this Technology age the amount of information available on the Internet and through other digital media has been increasing rapidly over recent years [1]. Due to this, customers are suffering to choose from the immense amount of available information and they have to put a lot of effort into searching and filtering information. Because of these challenges companies began studying and implementing helper systems that help customers in choosing and recommending items or goods based on their preference; which is called recommendation system. The recommendation systems focus mainly on recommending items and goods for customers based on prioritization and personalization by filtering and prioritizing immense amounts. [On the Internet, where the number of choices is overwhelming, items need to be prioritized and relevant information provided efficiently to make selection of available data easier](#) [2]. They reduce transaction costs of finding and selecting items in an online shopping environment. Recommendation systems have also proven to be in need of improvement for the decision-making process and quality [3] [4] and are therefore advantageous for both service providers and customers. There have been so many attempts and researches to create a recommendation system that recommends effectively by understanding customers' interests and priorities [1], [5]. Researchers suggest different techniques and mechanisms that recommend goods to customers. The most well-known mechanisms are content based recommendation, collaborative recommendation and hybrid recommendation. The content-based recommendation mechanism uses a number of discrete features or attributes of an element to recommend additional elements with similar elements properties [5], [6]. Collaborative recommendation uses user's previous behavior i.e. Previously purchased items and / or numerical ratings of these items and similar decisions by other users [5]–[7]. By combining content based and collaborative recommendation mechanisms a hybrid recommendation mechanism also can be created [6], [8].

Most of the recommender mechanisms have a problem of recommending a new item or product (without any rating) to customers and also have a problem of recommending items to the new user (without any history of rating an item). This problem is called the cold-start problem (when a system lacks the initial rating). In addition to the cold - start problem, another challenge of

researchers is scalability; that is, as more users and more items are added to the system, exactly how the system will recommend to the customer. And also, improper conversion of categorical dataset to numerical dataset will lead to information loss and this affects the recommendation accuracy. These problems can mislead the recommendation system and therefore users are not satisfied with the recommended elements. In order to solve these problems different researchers, suggest different solutions; however, clustering algorithms solve these problems better than other types of algorithms. In this research, first Elbow method will be used to find appropriate K or number of clusters [9], [10], second a well-known K-mean clustering algorithm is used for clustering the data set by using the K number of clusters determined by the Elbow method. To resolve the problem of clustering algorithms i.e. Creating optimal number of clusters, bio-inspired Particle Swarm Optimization Algorithm (PSO) is used to create Hybrid Particle Swarm Optimization Algorithm (PSO) and K-means (KM) to produce quality clusters [11], [12]. Finally, the result of clusters from K-means clustering algorithm will be compared with the result of clusters from Hybrid Particle Swarm Optimization Algorithm (PSO) in terms of cluster quality. The proposed solution in this research will be experimented in the MovieLens (ml-latest-small) dataset [13]. The efficiency of the proposed work will be evaluated by evaluation metrics.

1.2. Motivation

Recommendation systems are now a new field of study that catches the attention of customer-centric companies; that want to give recommendation services to their customers based on personalization and prioritization. A huge company like Amazon, Google, Netflix and others are now interested in the technology and they are implementing it in the way it suits them. Recommendation systems are now becoming one of the most researchable fields, because companies and researchers are now struggling to find a technique or a way that allows them to recommend products that fulfill the interest of most of the customers. Most of the researchers in the recommendation system are now focusing on solving cold-start and scalability problems to produce a recommendation system that results in more personalized recommendations. Researchers are tried and trying to solve these problems by applying different techniques and hybrid architectures. Even if different techniques are tried and implemented in years, there is still a need for more personalized recommendation and also a better way to solve cold-start and scalability problems. There is also a problem of data conversion that results in data loss when

categorical and mixed data sets are represented by numerical data improperly. This paper takes this as an initiation in solving these problems by implementing an effective hybrid architecture by using the Elbow method for finding K, Particle Swarm optimization and K-mean clustering algorithm together for creating optimal number of clusters.

1.3. Statement of the Problem

Most recommender systems try to recommend products based on the knowledge they have about the customer and the product recommended. Using the profile and previous history of customers to a recommendation system is useful to understand the users' preference and recommend products that fulfill those preferences. However, when a new user or product is added to the system, without previous history and rating it is difficult to recommend effectively. This problem is known as the cold-start problem, which hinders the recommender system's ability to recommend products based on customer preference. Cold-start problems create a trust issue for the customer, so that they might stop using recommender systems for their day to day activities. Even if clustering algorithms (K-Mean and fuzzy c-mean) are efficient in solving cold-start problems and also scalability problems, they suffer from creating an optimal number of clusters (usefulness of the clusters). Since poor clustering is poor prediction accuracy it might bring user dissatisfaction and trust issues. And also, one of the problems of Euclidean distance calculation is it works only for numerical data, when there is categorical attribute in the dataset other distance calculation is advised. These and other problems in recommendation systems may hinder a significant boost of recommendation system technology to facilitate customer service, recommending products to users based on their preferences and also reducing transaction cost. By understanding the potential of recommendation system, this research paper tries to answer the following research questions:

- 1) How can the cold-start problem be solved by using clustering algorithms (K-mean algorithm)?
- 2) How can clustering algorithm (K-mean) problem of creating an optimal number of clusters be solved by using bio-inspired optimization algorithms (Particle Swarm Optimization)?
- 3) What is the result of the Hybrid (PSO-KM) architecture when evaluated by evaluation matrices?

1.4. Objectives

1.4.1. General Objective

The general objective of this research is to design and implement a hybrid recommendation system by using Particle Swarm Optimization Algorithm and K-mean clustering algorithm.

1.4.2. Specific Objective

The following are the specific objectives of this research:

- To pre-process the dataset [for quality data](#),
- To select features for training the model,
- To convert data to give attributes a proper weight,
- To clean data to remove missing value rows,
- To initialize K (number of clusters) by using Elbow method,
- To implement K-means Algorithm to solve cold-start problem,
- To implement Hybrid Particle Swarm Optimization and K-means Clustering Algorithm to create optimal clusters,
- To Evaluate the result of the hybrid algorithm (Hybrid Particle Swarm Optimization and K-means Clustering Algorithm) regarding recommendation accuracy and efficiency.

1.5. Data Collection and Analysis

A data collection mechanism that will be used in this research is reading different papers that are downloaded from well-known research websites, such as the ACM Digital Library, IEEE Xplore, ScienceDirect and SpringerLink. The data set used to train the clustering algorithm downloaded from the internet is MovieLens (ml-latest-small) dataset [13].

1.6. Methodology of the Study

Recommendation systems are essential for customers for their day to day activities to get relieved from the burden of choosing and get personalized recommendation from an immense amount of information available on the internet. Recommendation systems are one of the many fruits of the advancement of technology and most customer-centric companies are now implementing this technology to reduce transaction cost, to facilitate selling and buying process and to fulfill

customer needs by recommending products based on their preferences. This research paper tries to design a model that solves the previously described problems and uses Hybrid Particle Swarm Optimization and K-mean clustering algorithm. For this research MovieLens (ml-latest-small) dataset is used and the attributes selected from this dataset for clustering are user_id, movie_id, genres and user rated movies or rating features. First, before the training begins the dataset passes through the pre-processing stage to produce quality data. Then, the clustering algorithm (K-mean) is applied to MovieLens (ml-latest-small) dataset. The clustering algorithm produces clusters that are categorized based on their similarity [14], [15]. Since K-mean clustering algorithm have a problem of creating an optimal number of clusters, this research paper uses a Hybrid Particle Swarm Optimization algorithm and K-means algorithm to produce optimal or homogenous clusters. First, by using the Elbow Method the number of cluster K is determined, then the PSO algorithm searches all space for a global solution [11], [12]. When the PSO algorithm finishes by finding the region of global optimum solution, then k-means algorithms start clustering to refine clusters to finalize the clustering task. After completing the Hybrid Hybrid Particle Swarm Optimization algorithm and K-means algorithm (PSO-KM), the research uses evaluation matrices to evaluate the work of the above hybrid architecture and K-means Clustering algorithms.

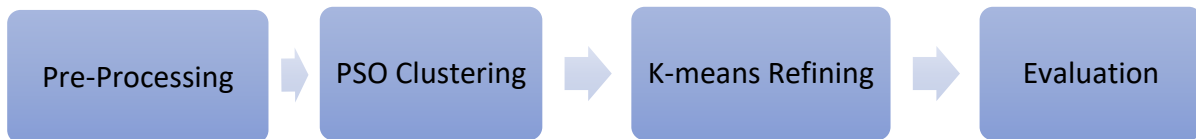


Figure 1.1 Work flow of the research

1.7. Scope and Limitations

Recommendation systems are one of today's research areas and researchers are trying to create a system that recommends products that are more personalized and prioritized based on customer preferences. This paper also considers these challenges and proposes a Hybrid Particle Swarm Optimization algorithm and K-means algorithm (PSO-KM) algorithm for recommendation accuracy. The following are the scope of this research: This research uses MovieLens (ml-latest-small) dataset for clustering and uses user_id, movie_id, genres and user rated movies or rating features attributes from the dataset. First, the dataset passes through the pre-processing stage to

produce numerical and quality data. Then the clustering algorithm (K-mean) is applied to MovieLens (ml-latest-small) dataset and the few possible clusters are produced. The pre-processed data can also be used to train the Hybrid Particle Swarm Optimization algorithm and K-means algorithm (PSO-KM) algorithm to produce quality clusters. Finally, the evaluation matrices are used to evaluate both K-means algorithm and Hybrid PSO-KM algorithm to see which algorithm produces quality clusters. Finally, the recommendation algorithm recommends movies based on their popularity.

As limitation of this paper is it only uses user_id, movie_id, genres and user rated movies or rating features attributes from MovieLens (ml-latest-small) dataset because of time and budget limitations. Considering other attributes will enhance the recommendation accuracy and also increase the effectiveness of the model or proposed solution. It has also been good if this hybrid architecture is tried and implemented in other data sets and by including more attributes.

1.8. Application of Results

The possible outcome of this research is the result of increased recommendation accuracy that satisfies most of the customer that uses the recommendation system. This research paper hopes it will show that the hybrid architecture (PSO-KM) will perform better in evaluation matrices than K-means Algorithm. The result of this research can be useful for customer-centric companies and for anyone who wishes to build recommendation systems. The following are the possible result of the research:

- The ability to recommend products for new users without the need of previous history.
- The ability to recommend a new item / product to a customer without the need of rating.
- The capacity of the system to handle the increased number of users and products (scalable).
- The improved recommendation accuracy by using hybrid architecture.

1.9. Thesis Organization

The research paper organized under seven chapters in the following ways:

Chapter One, discussed above, and It includes the introduction of the study, the motivation, and the statement of problems, the research questions that would be answered in the proposed solutions, the scope and limitation, the objective of the study, the methodology, and the application results of the study.

Chapter Two, presents the literature review and related works on [recommendation systems](#), Machine learning techniques, pre-processing techniques, hybrid techniques clustering algorithms, optimization algorithms, Movie-recommendation algorithms, and it discuss the related work.

Chapter three, discusses the research methodology used in this research work, dataset used to train the model, K-means clustering algorithms, Hybrid PSO-Km algorithm, and finally, it presents the evaluation methods.

Chapter four, discusses the proposed solution for solving cold-start problem and creating optimal clusters, i.e. Hybrid PSO-KM algorithm, and finally discusses popularity-based recommendation.

Chapter five, discusses the implementation and experimentation of the proposed solution. Including the working environment, dataset description, also the implementation of pre-processing, models implementations, and evaluation models.

Chapter six, discusses the result of the dataset pre-processing stage, the result of both K-means and Hybrid PSO-KM algorithm and also discusses the major results obtained by comparing both algorithms based on quality of clusters.

Chapter seven, presents conclusion, recommendation and identifies future works for further research direction on this research topic

Chapter Two

2. Literature Review

2.1. Overview

The tremendous increase in volume of information throughout the internet causes a challenge for users to filter and get the information they want from the huge information available. Information Retrieval Systems such as Google helps users in searching and finding the information, they need by accepting keywords (texts); however, these information retrieval systems are unable to deliver personalized and prioritized information for users [6]. The need of recommending and predicting systems is higher than ever, because searching, filtering and choosing process becomes tiresome for users and companies are implementing helper or recommender systems to facilitate selling and buying process, to reduce transaction time, to deliver personalized recommended items for users and to increase user's satisfaction. The recommendation system uses different algorithms and techniques to generate personalized recommendations based on the type of the service they offer. Clustering is one of the types of algorithms used in recommendation systems to group similar items in one group based on some distance calculation [16]. Data conversion is necessary when the dataset at hand is a collection of categorical and numerical data; and the conversion is done from one data type to another on a software level [17], [18]. Optimization algorithms are used in different fields to find the best possible/desirable solution [19]. Optimization algorithms are used to generate optimum recommendation results in combination with different algorithms. In this chapter, a few of the works and concepts of recommendation system, machine learning, clustering algorithm and optimization algorithm will be discussed.

2.2. Recommendation System

Recommender systems are information filtering systems that try to solve the problem of information overload for facilitating users with finding and buying of items [1]. The increase in digital information through the internet makes it difficult for users to identify items based on their interest through a wide variety of choices. Recommendation system also defined as any system that creates personalized recommendations as an output or guides to the user in a personalized way to interesting or interesting topics or

Useful objects in a large space of possible options [2]. In the mid-1990s studying Recommender systems for research purposes emerged independently and also recommending or predicting services for users began [1], [5]. From this time onwards, researchers tried and trying to find different ways and methods to develop systems that recommend items to users based on their interests and the type of item. Some of the recommender systems are: a GroupLens system for news article recommendation, the Ringo system for music recommendation, the famous Amazon e-commerce website and Letizia for recommendation on browsing the internet [1], [6]. The basic type of data needed for the recommender systems to recommend to others is the profile of users or background data which are the preferences the user has, input data which is the data that the user enters to the system for recommendation and also the algorithm that accepts the above information and recommends personalized and ranked objects [2]. Recommender systems help companies by reducing the transaction cost of filtering items in e-commerce systems and also it helps users by assisting their decision-making ability and prioritizing and personalizing items based on their interest [3], [20]. The recommendation system recommends or suggests to users' different choices to assist their decision-making process; these are: what items to buy, what music to listen to or whatever movie to watch too. Recommender system's role in giving suggestions to the user depends on the type of company or service provider service and the type of users. Recommender systems can be implemented in different areas such as: in tourism to increase the number of visitors in the destinations, in bookshops to increase the buyers' number, in different online shopping to increase user's satisfaction and reduce transaction cost. Companies aim to use a recommender system because of different reasons: to increase the number of items sold, to increase user's satisfaction, to sell more diverse items, to increase more user fidelity and to better understand what the user wants [5].

2.3. Phases of Recommendation system

The recommendation system mainly tries to suggest or recommend different items to users by studying their behavior or preferences and observing their web activities. In order to recommend accurately the recommendation system must pass through these major phases. The phases are: Information collection phase, Learning phase and recommendation phase. The following figure shows phases of recommendation system:

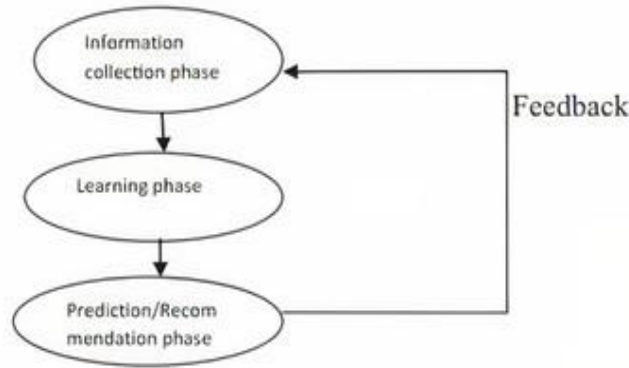


Figure 2.1 Recommendation phases [21]

2.3.1. Information Collection Phase

In this phase the required information about the user will be gathered to produce a user model or profile to recommend accurately. This information can be user attributes or user profiles and different items the user accesses. After all the necessary data about users is collected, the user model or profile will be completed. For recommending accurate suggestions for the user the recommendation agent needs all the possible information about the user. The recommender system focuses on three types of collecting information from the user. These methods are: explicit feedback which is the rating given by user or explicit input the user gives to the items based on his interest towards the item, implicit feedback which is preferences of the user gathered indirectly by observing user behavior and hybrid feedback which is the combination of the above two feedbacks. For the recommendation system to recommend and suggest the user accurate and reliable recommendations the system must have the user interest or user profile information in detail [6], [22].

2.3.1.1. Explicit Feedback

For better understanding of a user's interests and preferences the system creates a mechanism to accept user's feedback about the item he/she experienced directly or it is called user rating. This type of information is difficult to get because it needs the full cooperation of the user; most times. And also, it is the most reliable and accurate form of information, because it involves user's directly [6], [23].

2.3.1.2. Implicit Feedback

This kind of information is gathered by following up the user's behaviors and actions such as purchase history, browsing history, search patterns, content of the e-mail, or even mouse movements or clicks. In this information gathering process once the user gives approval for gathering usage data, no additional involvement is required by the user part. Since it doesn't require user involvement as in explicit feedback, this method is considered less accurate regarding the quality of data. However, when information gathered from explicit feedback is small in amount, implicit feedback came in handy [6], [24].

2.3.1.3. Hybrid Feedback

The explicit feedback and implicit feedback can be improved by combining the two information gathering methods. This combination will strengthen their strong part and also minimize their weaknesses [6], [22].

2.3.2. Learning Phase

Learning is distinctive attributes of intelligent behavior or it can also be defined as the modification of a behavior tendency, according to experiences which have been acquired through time [25]. The information about users gathered from the requirement gathering phase will be consumed by learning algorithms to filter and exploit the user features [6].

2.3.3. Prediction or Recommendation Phase

After all the necessary information about the user is gathered by different mechanisms the learning algorithm will be trained by this information or user profile, then finally the recommender system suggests or recommends items that the user might like by prioritizing items based on user interest. This recommendation is based on the type and quality of information gathered which can be explicit or implicit feedback and the type of learning algorithm used to train the information gathered [6].

2.4. Methods of recommendation system

Nowadays because of the exploding increase in the amount of information through the internet the recommender system or helper system is in need of helping users in their decision-making process in choosing items based on their interest. For an accurate and reliable recommendation for users

the recommendation system must use efficient and effective recommendation techniques based on the type of service at hand. The following figure shows the major recommendation techniques and also discussed below:

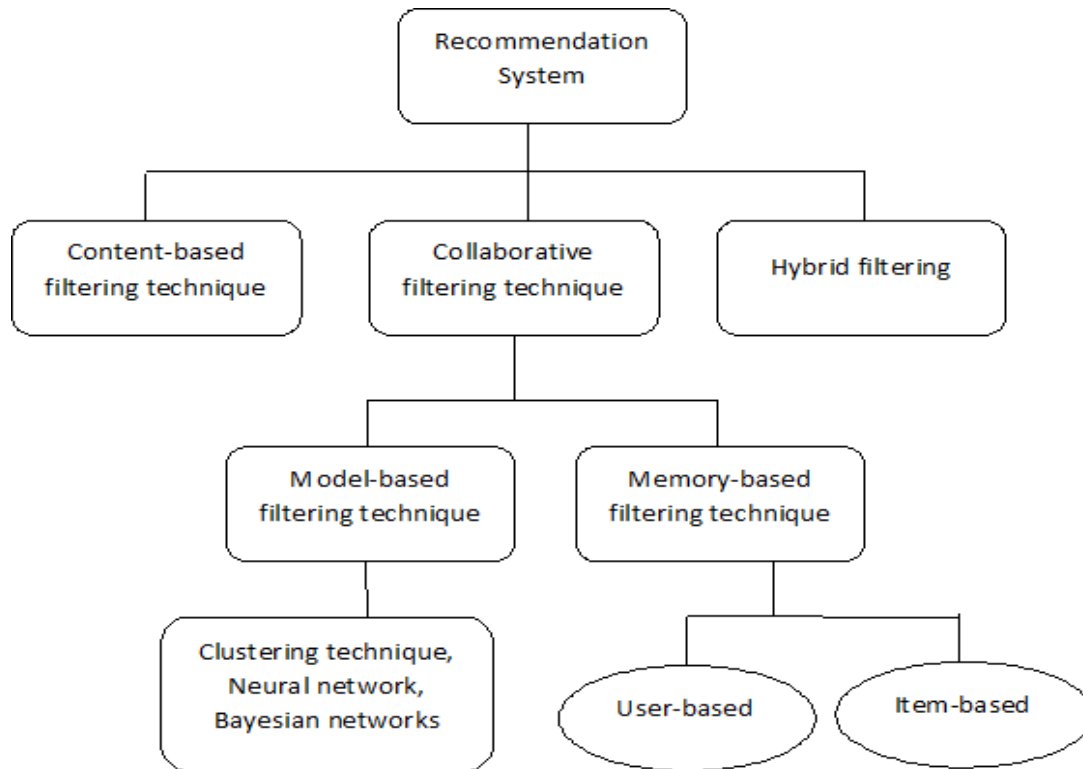


Figure 2.2. Recommendation Techniques [6]

2.4.1. Content-based Filtering

Content based filtering recommends based the user preferences using features extracted from the profile of the item that the user previously preferred [5]– [7]. The main assumption in content-based filtering is that by observing the past actions and histories of the user will allow us to predict the desired items in the future [25]. In this filtering technique unlike collaborative filtering techniques it doesn't need the predictions of other users, it predicts its recommendation by using only a given user profile or preferences. The basic operations on content-based filtering are: first, the profiles of items that are preferred by a given user are studied to come up with common properties that can be used to predict new items and stored as user preferences or profile, second, by comparing items with a given user profile items with a high degree of similarity are recommended [7]. The main advantage of this filtering algorithm is that they can recommend items to users, even if there is a shortage of information like ratings given by users. For accurate and

reliable recommendation this technique requires detailed knowledge of features of items in the profile [6]. In this filtering technique two major techniques are used to generate recommendation: the first technique uses traditional information retrieval methods, such as cosine similarity which can be used to measure the cohesion within clusters , the second technique uses statistical learning and machine learning methods to generate recommendations based on users interest by learning from user previous actions [7].

2.4.2. Collaborative Filtering

This recommendation technique recommends item to a given user that this given user didn't experience the item before based on the interests of users that have similar interests or liked or rated the item the same in the past [5]– [7], [26]. Since this recommendation is based on the interests of other users who have liked or preferred the same item in the past, it doesn't need deep knowledge of a user profile or item profiles like content-based filtering technique [26]. The collaborative filtering technique is good for recommending contents like movies and music that are difficult to describe by metadata [6]. The main assumption of this technique is that users who have the same interest in the past will have the same interest. The following diagram shows the collaborative filtering concept:

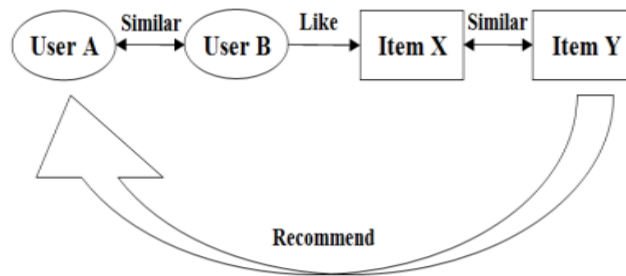


Figure 2.3 Collaborative Filtering [27]

There are two categories of collaborative filtering techniques, these are:

2.4.2.1. Model-Based Technique

Model-based technique utilizes previous ratings of the user to train a predefined model for improving recommendation accuracy [6], [28]. The use of pre-computed models allows this technique to recommend items to users very quickly [6]. The major machine learning algorithms

used in model building process are: Bayesian Classifier, Clustering, Regression, Decision Tree and Artificial Neural network.

Bayesian Classifier

This technique is a probabilistic framework that tries to solve classification problems by using conditional probability and Bayes theorem [6]. It uses previous knowledge and activities to predict items in the future.

Clustering

This technique tries to create groups or sub clusters from the collection of items based on their similarity by using some distance calculation [6], [16]. It is an unsupervised learning technique in which the subset in the created groups or sub clusters have a high degree of similarities and the subsets in different groups are dissimilar [29].

Regression

Regression techniques, mostly used when more than one item have linear relationship and also used to compare analytical methods and to test the biases of one method relative to another [6], [30]. It is usually used in curve fitting, prediction and testing relationship between variables systematically [6].

Decision Tree

This technique is a supervised learning technique that uses a tree graph which is constructed by analyzing a set of training examples for which the class labels are known [6], [31]. The quality of recommendation this technique recommends is dependent on the quality of data this technique trained on [6].

Artificial Neural Network

Artificial Neural Network (ANN) consists of many interconnected structural elements (neurons or nodes) that are arranged linearly and capable of performing parallel computing [6], [32], [33]. The idea of ANN is not to replicate the function of neurons, but to mirror the effective information processing abilities of the biological system [32]. Because of the ability to contain many neurons and the weights between each neuron connection, ANN is robust to noisy datasets and can handle complex relationships in datasets [6].

2.4.2.2. Memory-Based Technique

Memory-based algorithms use the entire user-item database (user-item matrix) of all users' preferences for all items and computes statistical computation for each prediction [34]–[36]. This technique performs the computation to find a set of users' or neighbors that have either similar ratings towards some items or bought similar items in the past [23]. After like-minded users or neighbors are selected for a given user different algorithms are used to combine preferences and produce suggestions or predictions of top N items for the given user [31]. The main advantage of this technique is that it is simpler to work with and new data can be added incrementally, but when the size of database or user-item matrix grows it becomes computationally intensive [35]. Memory-based technique can be implemented in two ways, these are:

I. Item-based Collaborative Filtering

This technique recommends to users based on the similarity of Items that the user rated or bought before [6]. By gathering all the items, the given user rated before and computes the similarity between gathered items and the given item, then k similar items are selected. After similar elements have been selected and their corresponding similarities have been calculated, the recommendation is based on a weighted average of the specified elements user's ratings on similar items [36].

II. User-based Collaborative Filtering

This technique recommends items to users based on the user ratings and other like-minded users' likings and ratings [36]. First, the similarity between users will be computed by using their rating towards the same item. Second, the predicted rating of the item by an active user is calculated as a weighted average of the ratings of the item by users who are similar to that given user [6].

III. User-Item matrix based Collaborative Filtering

This technique recommends items to users based on user-item (U-I) matrix, which encodes the individual preferences of users for items in a collection, for recommender systems [6]. First, this technique requires data that contains a set of items and a set of users who have reacted or rated some of the items. The rating or reaction can be on a scale of 1 to 5. The data in this technique is in the form of a matrix consisting of the rating given by a set of users to some items from a set of items [5,36]. Each row would contain the ratings given by a user, and each column would contain the ratings received by an item.

2.4.3. Hybrid Technique

This technique uses multiple traditional recommendation techniques described above, for better recommendation accuracy and to resolve the drawback of individual techniques [6]–[8]. The general assumption behind hybrid techniques is that by combining algorithms we can produce accurate and effective recommendations and also the limitations of individual algorithms will be resolved by combining them [6]. There seven different ways of combining individual recommendation techniques to build hybrids [8], these are:

i. Weighted Hybridization

Weighted hybridization combines the recommendation result of different recommendation algorithms numerically and by using a linear formula by taking the score from each technique will be combined to produce recommendation results [6], [8]. The merit of this technique can be the full utilization of recommendation techniques during the process of recommendation [6].

ii. Switching Hybridization

This technique tries to combine recommendation techniques by switching between techniques to avoid problems that are specific to a specific technique [6]. Based on the recommendation situation or the problem at hand the system chooses the technique based on its switching criteria [8]. The merits of this technique are that it has switching criteria to sense and switch based on its recommender; but because of switching criteria it introduces processing complexity [6].

iii. Cascade Hybridization

Cascade hybridization refines the result of one recommendation technique with another recommendation technique iteratively [6]. The strictly hierarchical and order dependence characteristics of this technique capable of noise tolerance and can produce a refined recommendation result [6], [8].

iv. Mixed Hybridization

In this technique the result of different recommendation techniques is presented side by side at the same time without mixing it [6], [8]. The challenging question in this technique is that if the results of the recommendation techniques are about to be merged, how the order of items are computed [8]? still not answered.

v. Feature-combination Hybridization

The feature-combination technique works by injecting the features produced by one technique such as content-based to another technique such as collaborative filtering [6], [8]. The merits of this technique can be its ability not always to depend on one technique [6].

vi. Feature-augmentation Hybridization

This technique requires additional features in addition to user rating and other information produced by other techniques [6]. Because the flexibility and the nature to add an additional feature feature-augmentation technique is more preferred than feature-combination technique [8].

vii. Meta Level Hybridization

In meta level technique one technique can use the internal model produced by another technique [6], [8]. The advantage of using this technique is that the models have more information than user ratings [6].

2.4.4. Knowledge-based Technique

Knowledge-based recommendation technique recommends items to users based on the detailed knowledge of users, items and their relationship to recommend items to meet the user's requirements [5], [8], [37]. This technique uses similarity function for comparing how much users need or preferences match recommended items [5].

2.5. Data Pre-processing

Data pre-processing is an important step of data mining and machine learning techniques that involves transforming raw data into an understandable format [38], [39]. Unprocessed raw data is often incomplete, inconsistent and / or has certain behaviors or trends and is therefore likely to contain many errors data cannot be sent through a model [38], [40]. Pre-processing a way to address these issues and send processed data to the model. The major tasks done under pre-processing are:

- **Data Cleaning:** Data cleansing data is cleaned up by processes such as filling in missing values to smooth out the noisy data, or resolving the data inconsistencies.
- **Data Integration:** Data with different representations are compiled and conflicts within the data are resolved.

- **Data Transformation:** Data is aggregated and generalized in a normalized manner.
- **Data Reduction:** This step aims to present a reduced representation of the data in a data warehouse.
- **Data Discretization:** involves reducing a number of values of a continuous attribute by dividing the range of attribute intervals.

2.5.1. Data Conversion Mechanisms

A formal definition of Data Conversion is that it is the conversion of one data format into another format at software level [17], [18]. In this context data conversion implies giving proper numerical value of categorical data to compute different distance calculations. Data conversion is usually used in machine learning to give the dataset attributes a proper weight when the datasets have mixed (numerical and categorical attributes) or categorical attributes. There are three forms of datasets depending on the type of attributes they have, these are pure numerical, pure categorical and mixed containing both numerical and categorical attributes. Categorical dataset values can be one of the following: continuous, ordinal and nominal. Some training algorithms that calculate intra-cluster distance using the Euclidean distance, such as k-mean works effectively only on numeric data [41]. If the dataset used to train the training algorithm is categorical or mixed, then either another distance calculation algorithm that uses other than Euclidean distance calculation is used as training algorithm or data conversion algorithm is used to give categorical attributes a proper weight without any information loss. Converting nominal or mixed data to numeric data is called encoding or continuation. There are different types encoding or continuation techniques, the following are techniques that are implemented in scikit-learn library, these are: Ordinal, One-Hot, Binary Helmert contrast Sum contrast Polynomial contrast Backward difference Contrast Hashing bases Leaveoneout and target coding [42]. And other methods or techniques used as distance calculation in case of mixed or categorical datasets are discussed below. K-harmonic mean type clustering [43] is used to overcome cluster center initialization by producing stable cluster centers and for categorical or mixed dataset cluster centers are represented by proportional distribution of categorical values in a cluster. The improved clustering algorithm for k prototypes [44] uses the distribution focus to represent the prototype of categorical attributes in a cluster and then the integrated mean which is the focus of distribution is to propose a new measure of dissimilarity to analyze the dissimilarity between data objects and prototypes. Gower's similarity coefficients [45]

is the simplest measure of similarity used to produce stable and satisfactory results for categorical dataset and has the flexibility and the ability to compare two different attribute types like numerical and categorical attributes. To handle datasets other than numerical type different researches [45]–[47] reveals that Gower’s similarity coefficients produce better results regarding categorical or mixed datasets.

2.6. Machine learning

Machine learning is defined as the automated detection of meaningful patterns in data [48]. It is also defined as the ability to predict the future based on the past [49]. Due to the nature of some problems or due to the fact that digital information is increasing in tremendous amounts, algorithms that can learn and adapt the patterns from huge amounts of information are needed. The learning process is the process of acquiring knowledge by using the experience and the training algorithm uses the training data to train and analyze the patterns of data. The two main problems that require machine learning algorithms because of its capability to learn and improve from experience are, tasks that are too complex to program (tasks that are performed by animals or humans and tasks that are beyond human capabilities) and the other problem is adaptability which requires the ability to adapt to user data [48]. Machine learning algorithms find the patterns in the data to better understand the data and help users in decision making in their day to day activities. Machine learning algorithms types often categorized into three, these are:

2.6.1. Supervised Machine Learning

The supervised machine learning algorithm trains a model to propose recommendations or suggestions for new items based on a known input dataset and known output dataset [50]. The training algorithm trained on the input data can produce a recommendation to a new item based on a known output dataset [51]. For producing a model supervised learning uses classification which is used in case of discrete outputs and regression which is used in case of continuous output.

2.6.2. Unsupervised Machine Learning

The unsupervised machine learning is used to train a model using a dataset that only has inputs and analyze to make a sense and to structure or group the data [50], [51]. It uses data that are unlabeled and uncategorized to recommend to users by creating some structure to the data to group.

It draws inferences from the dataset by using input data without output data. The most known unsupervised learning technique that is used for different applications is clustering algorithm.

2.6.3. Semi-supervised Machine Learning

Semi-supervised learning algorithms try to train the model by using data that contain a small amount of the data with output data and large amount of the data without output data (unlabeled data) [52]. In the case of data that contains small amounts of data with output data and large amounts of data without output data semi-supervised learning produces considerable excel in learning accuracy.

2.7. Clustering in Recommendation System

Clustering algorithm is an unsupervised learning technique that aims to partition a given dataset item into homogenous groups or clusters, in which items in the same cluster are similar and items in different clusters are dissimilar [46], [47], [53]. It is called unsupervised learning because it uses dataset that are unlabeled or without output data and it is the clustering algorithm's responsibility to analyze and make sense of the data to find the pattern to categorize them into similar groups or clusters. The final clusters created might be different in density, shape and size because of the type of clustering algorithm used, the nature of the dataset and the number of clusters k . The application of clustering algorithms can be used in a broad variety of fields a few are: pattern recognition, image processing, machine learning, statistics, mining text data, sentiment analysis, profiling, spatial databases and web mining [53]. The following diagram shows how cluster are created:

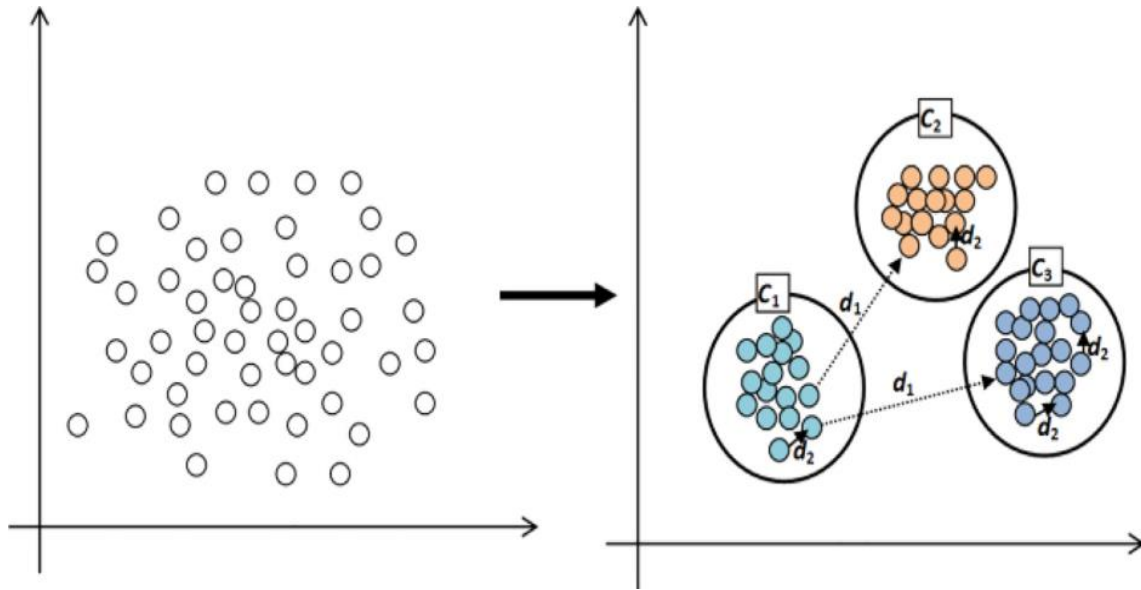


Figure 2.4 Cluster Formation [53]

There are different clustering methods that are used for different purposes, these are:

2.7.1. Partitional Method

In this clustering, to create a set of disjoint clusters fixed number of k data points are chosen to function as centroid [53], [54]. After the centroid selection the distance is calculated between each element and the centroid, then the items will be assigned to the closest cluster centroid based on the calculation. After the reassignment of items in all clusters, a new centroid is calculated and again the distance is calculated between each item and the new centroid in each cluster, and again the items are reassigned to the closest cluster centroid and this will continue until all items are in their closest cluster. This clustering method creates clusters by assigning items to closest clusters by fulfilling the following two requirements; the first requirement is each cluster contains at least one centroid point and the second requirement is each centroid point belongs to exactly one cluster [54]. A lot of algorithms are introduced in different literatures using different type of centroids for clustering depending on the type dataset they used [53]:

K-mean

K means that Macqueen 1967 is one of the simplest unattended learning algorithms that solve the well-known clustering problem. K-means Clustering is a method of vector quantization that originally comes from signal processing and is popular for cluster analysis in data mining.

k-mean clustering algorithm is one of the most efficient and most widely used algorithms for solving different problems in different topics because of its simplicity and low complexity [16]. It is an iterative algorithm that tries to create homogenous clusters by representing each cluster by mean or weighted mean known as centroid [46]. First, the user chooses k centroid points to start with as reference points and then the items are clustered into k points based on the closeness of the item to cluster centroid coordinate [46]. To create clusters that contain homogenous items this algorithm uses a Euclidean distance calculation to find the correct centroids and reassign items in their closest clusters. The Euclidean distance calculation for finding new centroid points and the reassignment of items to the closest cluster is performed iteratively until all elements are in their closest cluster [44], [46]. The main advantages of k-mean algorithms are: simple, fast, relatively flexible, highly efficient and it can effectively represent only numerical data with geometrical and statistical meaning. The main disadvantages of k-mean algorithm are: sensitivity to initial k (the number of clusters) values, works effectively only with numerical data, highly dependent on the initial cluster centers and due to its iterative nature, it is not applicable if the data is too large [44].

K-modes

K-modes algorithm was proposed by Huang (1998) and it is the modified version of k-mean algorithm that tries to represent by using a simple matching dissimilarity measure and instead of mean it uses modes. To minimize cluster costs, the frequency-based method is used to update the modes in the cluster process with the appropriate degree of similarity for categorical variables. The modification made to the k-means algorithms enhances k-modes algorithm to efficiently cluster large categorical datasets [55].

K-medoids

K-medoids clustering algorithm is a variance of k-mean algorithm, instead of using mean point as a centroid, k-medoids use an actual point in the cluster as a centroid. Since using the mean as centroid is greatly influenced by the outlier, thus a correct representation of the cluster center cannot happen. The problem of this algorithm is it is not scalable enough for large datasets [56].

2.7.2. Hierarchical Method

Hierarchical clustering is a type of clustering that tries iteratively to create the hierarchy of clusters without specifying the initial number of clusters. The strategies to create hierarchical clusters can be divided into two. The first is Agglomerative clustering that uses a bottom-up approach for creating hierarchical clusters and as one moves up individual clusters with lowest distances between corresponding cluster centroids are merged. To minimize cluster costs, the frequency-based method is used to update the modes in the cluster process with the appropriate degree of similarity for categorical data of the clusters are performed as one goes down the hierarchy. The main demerit of this algorithm is it has relatively high time complexity. The potential of this algorithm can be utilized if it is used for clustering arbitrary attribute type and data shape. The following algorithms fall in this category, these are: BIRCH (Balance Iterative and Clustering using Hierarchies), ROCK (Robust Clustering using Links) and CURE (Clustering using Representatives) [53].

2.7.3. Density-based Method

In a density-based clustering algorithm, clusters are selected based on the density of the points in the dataset and clusters are considered a high-density region [46], [53]. This algorithm can better deal with large datasets with noise and also can select and cluster points with different size and shape. The main demerits of this algorithm are in dealing with high-dimensional datasets and having high time complexity. The main algorithms that fall in this category are: DBSCAN (Density-Based Spatial Clustering of Applications with Noise), OPTICS (Ordering Points to Identify the Clustering Structure) and SNN (Similar Nearest Neighbors) [53].

2.7.4. Grid-based Method

In grid-based clustering, the original dataset is divided into a finite number of cells to create clusters that have dense grid cells. A cell can be a region, a cube and a hyper rectangle [46]. Most algorithms that are grid-based require the specification of density thresholds or the grid size. Algorithms that fall in this category are: STING (Statistical Information Grid) and WaveCluster [53].

2.8. Optimization Algorithms

Optimization is applicable in any field, it helps to find the best possible/desirable solution and the objective of optimization can be different, it might be to optimize energy consumption, cost, increase profit, output, performance and efficiency [19], [57]. In most cases, we use optimization to optimize limited resources such as time and money in real world applications to use them optimally under various constraints. Because of enormous computational efforts they take the previous methods used to solve optimization problems tend to fail as the problem size increases [19]. For this reason, bio-inspired stochastic optimization algorithms are inspired to use computationally efficient alternatives to deterministic approaches.

Nature inspired algorithms are algorithms that are inspired by nature and that try to imitate the ways of nature to solve very complex problems. Nature has simple and efficient ways of solving complex problems using very simple initial conditions and rules that appear very complex in computing. Because of nature's ability to find optimal solutions for all the phenomena and problems that are encountered, researchers are studying to mimic all the mysterious and deep capabilities of nature to solve complex optimization problems. Bio-inspired computing has come up as a new field in computing that tries to model and mimic nature's problem-solving capacity to improve the ability of computers to solve problems and give optimum solutions. The two most successful classes of Biological Inspired Algorithms are evolutionary algorithms which are inspired by the natural evolution and swarm-based algorithms Inspired by collective behavior in animals [19].

Swarm Intelligence is an emerging study area in bio-inspired computing developed by Kennedy and Eberhart in 2001 for implementing adaptive systems [19]. Swarm intelligence is based on the collective social behavior of organisms, but evolutionary algorithms are based on genetic adaptation and natural evolution of organisms. Some of the algorithms that are categorized in this paradigm are[19]: Particle swarm optimization (PSO) is proposed by Kennedy and Eberhart in 1995 and inspired by the social behavior of bird flocking searching for food, Ant Colony Optimization (ACO) is proposed by Dorigo & Di Caro in 1999 and inspired by the foraging behavior of ants, Artificial Bee Colony Algorithm (ABC) is proposed by Karaboga and Basturk and inspired by intelligent foraging behavior of a honeybee, Artificial Algae Algorithm (AAA) [58] is proposed by Sait Ali Uymaz, Gulay Tezel, Esra Yel in 2015 and inspired by the living

behaviors of microalgae (which is based on evolutionary process, adaptation process and the movement of microalgae).

Particle Swarm Optimization (PSO) is one of evolutionary and population-based search algorithms which simulates the movement and flocking of birds [15], [59], [60]. The birds in a flock are symbolically represented as particles, which is initialized with a population of random solutions. The algorithm maintains a population of particles which seems as simple agents moving through a problem space, where each particle represents a potential solution to an optimization problem [11], [12]. The position of a particle in the multi-dimensional problem or function search space is a possible solution to the problem. Each particle has memory and flies with a certain velocity and also moves in problem space to find the global best position in an iterative process. At each iteration of the algorithm, the velocity vector for each particle is modified based on three conditions: the particle momentum, the best position reached by the particle and that of all particles up to the current stage [59]. In the next iteration, the particle is moved to its next position based on the determined velocity for each particle. When a particle moves to a new position or location, a different solution for the problem is generated. Then this solution is evaluated by a fitness function to provide a quantitative value of the solution's utility. The PSO algorithm consists of three general steps, these are: generation of particles (generating particle's positions) and their information (velocities), movements (velocity update) and new information vector (position update) [15]. PSO is one of an evolutionary computation technique that offers globalized search methodology which helps to get a more precise clustering efficiency.

In 2007 Kyoung-jae K. and Hyunchul A. proposed a recommendation system using GA (generic optimization algorithm) K-means clustering in online shopping market to produce relevant clusters more effectively and to effectively segment the online shopping market [61]. In 2012 Chia-Cheng H., Hsin-Chin C. et al. uses an artificial bee colony optimization algorithm (ABC) to a personalized auxiliary material recommendation system based on learning style on Facebook to produce a personalized auxiliary material recommendation system based on the degree of difficulty of the auxiliary materials, individual learning styles, and the specific course topics and ABC optimizes the system and improved the execution time and accuracy of the recommendation system in an efficient manner [62]. In 2016 Sumathi. G, Sendhilkumar. S and Mahalakshmi. G.S used Particle Swarm Optimization algorithm to produce relevant pages for users and by using PSO the proposed system solves cold-start problem also it effectively enhanced the quality of recommendation

system with the better precision, recall and F-Score values [63]. In 2016 Xibin W., Fengji L. et al. used improved Particle Swarm Optimization algorithm to A Personalized Electronic Movie Recommendation System with the use of both support vector machine (SVM) and PSO the recommendation system produced a movie recommendation list by including the movie's content information, users' demographic and behavioral information to better capture the users' interests and preferences efficiently [64].

2.9. Movie Recommendation System

In this technology age, one of the major benefits of the internet is that it revolutionized the way information is collected, stored, processed, presented, shared and used [65]. Due to the ever-increasing amount of information on the internet, it becomes difficult to deal with a lot of information and it is also difficult for users to choose from a lot of choices. This condition is known as Information overload. By using new technologies like recommendation systems now it's possible for systems to take advantage of the enormous amount of information that is available on the internet and to filter and prioritize items to recommend to users based on their interest. The information overload is not a problem only in case of research, it is also a problem in the entertainment business. Nowadays recommendation systems are developed to recommend users about web pages, movies, music, restaurants, news articles and tv programs by understanding users' interest. Recommending movies to users based on their interest is very difficult because of the nature of details or features of the movies, such as: country, genre, year, participants (directors, actor, script writers) and movie company. Websites like Netflix, are used by a huge amount of people to get personalized movie recommendations based on the user's ratings and other user ratings [66]. Researchers are still trying to find best recommendation system architecture by combining different algorithms that can recommend users a personalized and prioritized movie by using all information (profile, cookies, web pages visited, other users' ratings and etc.) available about users and movies.

2.10. Review Papers

Recommender systems are based on a variety of approaches such as content based [1], collaborative approach [2], hybrid [3,4]. Most film recommendation systems focus on collaborative filtering and clustering. In film recommendation systems, the user can rate the films

that the user has seen. These ratings are then applied to recommend other movies to the user that user has not seen by using collaborative filtering that is based on similar ratings. Collaborative filtering [5] is tremendously used and this approach used by most of the recommender systems. Clustering based algorithms are repeatedly used in movie recommendation systems to solve the problem of scalability. Many researchers applied clustering approaches on recommender systems that produced effective recommendations [6,7]. The purpose of the cluster is to divide objects into groups called clusters so that two objects within the same cluster have a minimum distance between them, to identify similar objects, the clustering process is carried out offline to create the model.

To overcome the limitations of a collaborative recommendation system, this article proposes a hybrid cluster and optimization technique to improve the accuracy of the movie prediction. The research motive is to design a model that utilizes attributes (user_id, movie_id, genres and user rated movies or rating features) from the MovieLens (ml-latest-small) dataset for predictions. It also aims at hybrid architecture to solve the very problems of recommendation systems i.e. cold-start and scalability. Hybrid PSO-KM Optimization algorithm and then apply to MovieLens (ml-latest-small) dataset for improved efficient recommender systems. Initially the k-mean clustering algorithm is applied to MovieLens (ml-latest-small) dataset for clustering of users into different clusters. K-means clustering uses centroids, K different randomly-initiated points in the data, and assigns every data point to the nearest centroid [14], [15]. After every item's point has been assigned, the centroid is moved to the average of all of the points assigned to it. Then the process repeats: every point is assigned to its nearest centroid; centroids are moved to the average of points assigned to it. The algorithm is done when no point changes assigned centroid. In this research, improving k-mean algorithm in a way to make it create the optimal number of clusters efficiently will be considered [59]. First, before the training begins the dataset passes through the pre-processing stage to produce quality data. Then, the clustering algorithm (K-mean) is applied to MovieLens (ml-latest-small) dataset. The clustering algorithm produces clusters that are categorized based on their similarity. Since K-mean clustering algorithms have a problem of creating an optimal number of clusters, this research paper uses a Hybrid Particle Swarm Optimization algorithm and K-means algorithm to produce optimal or homogenous clusters. First by using the Elbow Method the number of cluster K is determined, then the PSO algorithm searches all space for a global solution. When the PSO algorithm finishes by finding the region of

global optimum solution, then k-means algorithms start clustering to refine clusters to finalize the clustering task [11], [12], [15], [59]. After completing the Hybrid Hybrid Particle Swarm Optimization algorithm and K-means algorithm (PSO-KM), the research uses evaluation matrices to evaluate the work of the above hybrid architecture and K-means Clustering algorithms.

2.11. Summary

The problem of information overload is the problem for users by creating difficulty in searching and selecting items from the enormous amount of information available on the internet and for a company also it creates a difficulty for the buying and selling process by increasing transaction time for users, which decreases the possible amount of cost the company gets. To overcome the problem of information overload, the researchers are developing recommendation systems that recommend users by using concepts like prioritization and personalization. Different customer-centric companies are now implementing a recommender system in their core system to minimize the transaction cost and to increase users' satisfaction.

Researchers have proposed and propose different solutions to solve the well-known problems in recommendation systems that are hindering to recommend based on users' interest. These problems are like a cold - start which is the systems capability in which the system is unable to recommend new items to users and unable to recommend items to new users, scalability which is unable to add new items and new users without decreasing of performance and the ability of the system to process a huge amount of information and deliver quality recommendations to users. This topic is one new researchable topic and researchers are now tried and trying to propose different system architectures that can solve these and other problems that hinder to produce quality recommendations. The usage of algorithms and techniques depends on the type of data used, the type of service and the type of users.

Finally, to optimize different design errors in different fields optimization algorithms is used. The optimization algorithms can be used to optimize scarce resources for different purposes. These resources that need optimization can be: energy consumption, cost, increase profit, output, performance and efficiency. Nature inspired algorithms are a well-known optimization algorithm that are used to optimize resources in different fields by using nature's way of problem solving and optimizing process. In recommendation system, different nature inspired optimization algorithms are used in different literature to increase the efficiency, to minimize running time and

for quality recommendation results. The proper usage of optimization algorithm in hybrid architecture in recommendation system can empower the system to suggest quality recommendations.

Chapter Three

3. Research Methodology

The recommendation system this research paper tries to design uses Hybrid K-mean clustering algorithm and PSO algorithm (PSO-KM) to solve problems like cold-start, data conversion problem and creating an optimal number of clusters. For this research MovieLens (ml-latest-small) dataset is used and the attributes selected from this dataset for clustering are user_id, movie_id, genres and user rated movies. The main processes for this proposed hybrid recommendation architecture are: First, to clean the dataset for missing values, to remove unwanted columns and to convert categorical columns to numerical the dataset passes through the pre-processing stages. Then second, clustering algorithm (K-mean) is applied to MovieLens (ml-latest-small) dataset. The clustering algorithm produces clusters that are categorized based on their similarity. Since K-mean clustering algorithms have a shortage of creating an optimal number of clusters, this research paper uses a Hybrid Particle Swarm Optimization and K-mean algorithm (PSO-KM) to produce optimal number of clusters. Finally, the result of clusters from K-means clustering algorithm will be compared with the result of clusters from Hybrid Particle Swarm Optimization Algorithm (PSO) and K-means (PSO-K-means) in terms of cluster quality. Using this hybrid structure, i.e., PSO-KM enhances the prediction accuracy greatly. After completing the Hybrid PSO-KM algorithm, the research uses evaluation matrices to evaluate the work of the above hybrid architecture.

3.1. Data Collection Mechanism

In this stage of the research, a search of available information in the area of mainly recommendation systems, machine learning, dataset attribute data type conversion, clustering algorithm, optimization algorithm and evaluation matrices must contain a strategy that leads to the inclusion of relevant papers and exclusion of irrelevant papers. This search strategy requires querying reference databases with custom search strings and manually filtering query results using predefined databases inclusion and exclusion criteria.

3.1.1. Search Strategy

The selection of studies is governed by inclusion and exclusion criteria which should be based on the research questions. Papers will be included in the review if the paper describes research into the work and concepts of the recommendation system. Included papers that are published in various trade journals and conference reports from highly regarded digital portals like:

- IEEE
- ACM digital library
- Elsevier
- Springer
- Popular websites.

In this research, papers are included that are published between 2003 and 2018. January 2003 is chosen as a starting date, because as a researcher, the gap should not exceed more than fifteen years and December 2018 is chosen as the end date. The searching criteria for the review are prepared based on works and concepts of recommendation system and based on the inclusion and exclusion criteria predefined. All the items in our search string are connected with OR statements and the search strings used in this research are listed below:

- Recommendation system
- OR Machine learning
- OR Recommendation techniques OR Recommendation algorithms
- OR Clustering algorithms OR K-means clustering
- OR Categorical data conversion
- OR Optimization algorithms OR Particle Swarm Optimization
- OR Evaluation matrices OR Cluster Evaluation matrices

3.1.2. Inclusion and Exclusion Criteria

In this research inclusion and exclusion criteria for filtering papers are formulated. If a paper met an exclusion criterion, then the paper was removed from the study. Only if a paper met all inclusion criteria, the paper was kept.

The Inclusion Criteria

- The study refers to the recommendation systems.
- The paper focuses on the topic of Hybrid Recommendation systems.
- The area of the study should focus on hybrid recommender systems that use mixed dataset.

The Exclusion Criteria

- Papers in a language other than English.
- Papers published before 1st of January 2003.
- Papers downloaded other than IEEE, ACM, Elsevier, Springer and popular websites.
- Papers focused on other than recommendation systems or algorithms.
- Papers other than articles or journals and conference papers.

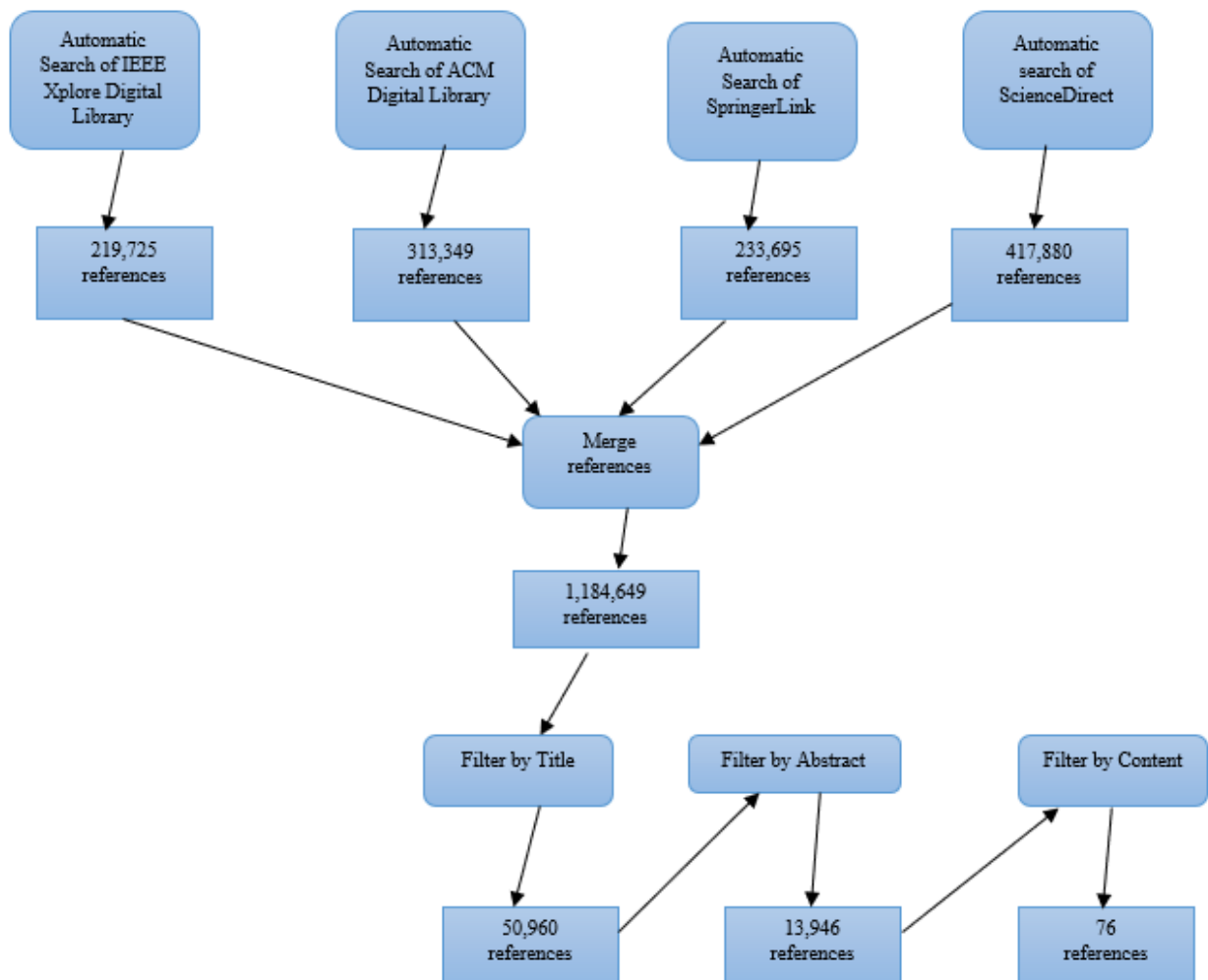


Figure 3.1 Literature review path

3.1.3. Dataset Used

The data set used for clustering is the MovieLens data set (ml-latest-small), known for its quality, which is used for so many examinations. The data set was created on September 26, 2018 by the GroupLens Research Group at the University of Minnesota [13]. This MovieLens data set (ml-latest-small) is a smaller version of the data sets prepared by the GroupLens Research Group. This data set (ml-latest-small) describes the 5-star rating and free text tagging activity of [MovieLens] (<http://movielens.org>), a film recommendation service. It contains 100836 ratings and 3683 tag applications for 9742 films. This data was compiled by 610 users between March 29, 1996 and September 24, 2018. This data set was created on September 26, 2018. The users were chosen at random for inclusion. All selected users had rated at least 20 films. No demographic information is included. Each user is represented by an ID and no further information is provided. The data is contained in the files "links.csv", "movies.csv", "reviews.csv" and "tags.csv". **Feature Selection Process:** it is a process in which the researcher automatically or manually selects those features which contribute most to the research prediction or recommendation variable or output in which the research is interested in. Using a feature selection process before training data can reduce overfitting, reduce training time and improve accuracy [67]. The three main techniques for feature selection are wrappers that use a predictive model to evaluate subsets of features, and each new subset is used to train a model, filters (select subsets of items independently of the used classifier as a pre-processing step) and embedded methods (includes group of techniques which perform feature selection as part of the model construction process) [68].

Due to time, budget, processing power and memory space constraints 'movies.csv', 'ratings.csv' files are selected to train the model. From these files' user_id, movie_id, genres and user rated movies or rating features are used to train the model. Other features from these files are deleted by using python drop-column functionality. The two independent files will be merged by python merge-tables functionality. After merging the number of tuples is 1,000,386.

Before the training the model process starts, all missing value rows or tuples should have to be cleaned. After all missing value rows are cleaned the number of tuples is 1,000,209. After the pre-processing stage is completed, the remaining columns are:

- MovieID -: unique identification number for each movie
- UserID -: unique identification number for each user

- Title -: title for each movie
- Geners -: movie type for each movie (a movie can have more than one genre)
- Rating -: users rank (1-5) given for each movie

Attribute/Data selection criteria

- select only a subset of the available data that accurately represent the entire population.
- select features that have a high influence on recommended results.
- features not related to the problem at hand can be excluded.

3.2. Review Papers

Recommender systems are based on a variety of approaches such as content based [1], collaborative approach [2], hybrid [3,4]. Most film recommendation systems focus on collaborative filtering and clustering. In film recommendation systems, the user can rate the films that the user has seen. These ratings are then applied to recommend other movies to the user that user has not seen by using collaborative filtering that is based on similar ratings. Collaborative filtering [5] is tremendously used and this approach used by most of the recommender systems. Clustering based algorithms are repeatedly used in movie recommendation systems to solve the problem of scalability. Many researchers applied clustering approaches on recommender systems that produced effective recommendations [6,7]. The purpose of the cluster is to divide objects into groups called clusters so that two objects within the same cluster have a minimum distance between them, to identify similar objects, the clustering process is carried out offline to create the model.

To overcome the limitations of a collaborative recommendation system, this article proposes a hybrid cluster and optimization technique to improve the accuracy of the movie prediction. The research motive is to design a model that utilizes attributes (user_id, movie_id, genres and user rated movies or rating features) from the MovieLens (ml-latest-small) dataset for predictions. It also aims at hybrid architecture to solve the very problems of recommendation systems i.e. cold-start and scalability. Hybrid PSO-KM Optimization algorithm and then apply to MovieLens (ml-latest-small) dataset for improved efficient recommender systems. Initially the k-mean clustering algorithm is applied to MovieLens (ml-latest-small) dataset for clustering of users into different clusters. K-means clustering uses centroids, K different randomly-initiated points in the data, and

assigns every data point to the nearest centroid [14], [15]. After every item's point has been assigned, the centroid is moved to the average of all of the points assigned to it. Then the process repeats: every point is assigned to its nearest centroid; centroids are moved to the average of points assigned to it. The algorithm is done when no point changes assigned centroid. In this research, improving k-mean algorithm in a way to make it create the optimal number of clusters efficiently will be considered [59]. First, before the training begins the dataset passes through the pre-processing stage to produce quality data. Then, the clustering algorithm (K-mean) is applied to MovieLens (ml-latest-small) dataset. The clustering algorithm produces clusters that are categorized based on their similarity. Since K-mean clustering algorithms have a problem of creating an optimal number of clusters, this research paper uses a Hybrid Particle Swarm Optimization algorithm and K-means algorithm to produce optimal or homogenous clusters. First by using the Elbow Method the number of cluster K is determined, then the PSO algorithm searches all space for a global solution. When the PSO algorithm finishes by finding the region of global optimum solution, then k-means algorithms start clustering to refine clusters to finalize the clustering task [11], [12], [15], [59]. After completing the Hybrid Particle Swarm Optimization algorithm and K-means algorithm (PSO-KM), the research uses evaluation matrices to evaluate the work of the above hybrid architecture and K-means Clustering algorithms.

3.3. Proposed solution or Framework

The proposed framework pre-processes the data to produce quality data that is numerical and no missing values that is ready for distance calculation. The k-means clustering algorithm which is used for creating clusters and by using PSO-K-means (PSO-KM) which is used for optimal clusters to solve the problem of K-means algorithm which is creating optimal number of clusters. And finally, the result of clusters created by K-means are compared with clusters created by PSO-K-means (PSO-KM). The following are the main parts of proposed hybrid recommendation architecture:

Data Pre-processing: Data pre-processing is an important step of data mining and machine learning techniques that involves changing/ transforming raw data into an understandable format [38], [39]. Unprocessed raw data is often incomplete, inconsistent and / or has certain behaviors or trends and is therefore likely to contain many errors, so this data cannot be sent through a model

[38], [40]. For better accuracy in recommendation results, data should have to pass through different steps of data pre-processing to clean data from inconsistent and missed values and to prepare the data in proper format for the filtering algorithm. The MovieLens (ml-latest-small) dataset passes through the following pre-processing stages, these are: feature selection, data cleaning, data integration and data conversion.

k-means Clustering Algorithm: it is one of the most efficient and most widely used algorithms for solving different problems in different topics because of its simplicity and low complexity [16]. It is an iterative algorithm that tries to create homogenous clusters by representing each cluster by mean or weighted mean known as centroid [46]. To create clusters that contain homogenous items k-means algorithm uses Euclidean distance calculation to find the correct centroids and reassign items in their closest clusters.

Hybrid Particle Swarm Optimization (PSO-KM) Algorithm: By combining the two algorithms, i.e. PSO-K-means (PSO-KM), first by using the Elbow Method the number of cluster K is determined, then the PSO algorithm searches all space for a global solution. When the PSO algorithm finishes by finding the region of global optimum solution, then k-means algorithms start clustering to refine clusters to finalize the clustering task. The ability of PSO algorithm to find global optimum solutions can be used in K-Mean algorithm to find the local optimal solution [15].

3.4. Evaluation Matrices of the Model

Clustering algorithms are more difficult to evaluate. Internal metrics only use information about the calculated clusters to assess whether the clusters are compact and well separated. External metrics also can be used to perform statistical tests on the structure of your data. Algorithms belonging to the family of unsupervised learning cannot predict variables linked to the data. Instead of having an output, the data has only one input, which is a set of variables that describe the data [69]. In general, cluster validation statistics can be divided into three classes:

Internal Cluster Validation: - which uses the cluster's internal information to evaluate the quality of a cluster structure without reference to external information. It can also be used to estimate the number of clusters and the appropriate clustering algorithm without external data

External Cluster Validation: - This consists in comparing the results of a cluster analysis with an externally known result, for example with class names provided externally. It measures the

extent to which cluster labels match externally provided class labels. Since we know the “true” cluster number in advance, this approach is mainly used to select the right clustering algorithm for a particular data set.

Relative Cluster Validation: - which evaluates the cluster structure by varying different parameter values for the same algorithm, e.g. By varying the number of clusters k, it is generally used to determine the optimal number of clusters.

The evaluation can be:

- The cluster quality of the K-means clustering algorithm.
- The cluster quality of the hybrid PSO-KM algorithm.

The following methods will be used for evaluating the model cluster results:

- 1) **Silhouette score:** Silhouette score tells how far away the data points in one cluster are, from the data points in another cluster. The range of silhouette score is from -1 to 1. Score should be closer to 1 than -1.

Silhouette analysis can be used to determine the degree of separation between clusters. For each sample the steps are:

- Compute the average distance from all data points in the same cluster (a_i).
- Compute the average distance from all data points in the closest or neighboring cluster (b_i).
- Compute the coefficient:

$$\frac{b^i - a^i}{\max(a^i, b^i)}$$

Equation 3.1 The computation of Silhouette Score

Since coefficient can only take values in the interval [-1, 1].

- If it is 0 → the sample is very close to the neighboring clusters.
- If it is 1 → the sample is far away from the neighboring clusters.
- If it is -1 → the sample is assigned to the wrong clusters.

- 2) **Clustering Criterion Function:** it can be defined that measure the overall clustering quality. Using different clustering criterion functions for determining their partitions can produce varying results when applied to identical data. The most commonly used clustering

criterion functions is the Sum-of Squares-Error (SSE) criterion. SSE is the total sum of the squared distance between all samples and its cluster center.

The SSE criterion function is given by:

$$E = \sum_{j=1}^K \sum_{x_i \in v_j} ||x_i - v_j||^2$$

Equation 3.2 The computation of SSE Criterion

where V_j is the mean value of samples in j_{th} cluster. The goal in SSE clustering is to obtain a partitioning of the data set, such that E is minimized. SSE criterion is valid for cluster sample dense as well as the small differences in the number of various clustering samples. But if the shape and size of the cluster varied greatly, use of the sum-of-squares error rule may cause error clustering.

3) Quantization Error

The fitness of algorithm is easily measured as the quantization error:

$$J_e = \frac{\sum_{j=1}^{N_c} [\sum_{\forall Z_p \in C_{ij}} d(Z_p, m_j) / |C_{ij}|]}{N_c}$$

Equation 3.3 The computation of quantization error

where d is the distance to the centroid, and $|C_{ij}|$ is the number of data vectors belonging to cluster C_{ij} i.e. the frequency of that cluster. The goal of quantization Error is to minimize the average (squared) quantization error.

Chapter Four

4. Proposed Hybrid Recommendation System

4.1. Proposed Hybrid Model

In this section the solutions to the problems listed in [section 1.3](#) are discussed in detail. This research paper tries to design a model that solves this research problem by using Hybrid K-mean clustering algorithm and Particle Swarm optimization (PSO) algorithm. First, pre-processing will be done on the MovieLens (ml-latest-small) dataset to resolve issues like inconsistency, missing values and noisy data to prepare the dataset properly for training algorithms. Second, after preprocessing is done on the dataset the clustering algorithm (K-mean) will be applied to MovieLens (ml-latest-small). The clustering algorithm produces clusters that are categorized based on their similarity. Third, since K-mean clustering algorithms have a problem of creating optimal number of clusters, this research paper uses Particle Swarm Optimization (PSO) algorithm to produce quality clusters. First, the k-means algorithm is used to create homogenous clusters. Second, the hybrid PSO-K-means (PSO-KM) algorithm is used to create homogenous clusters. Finally, the result of clusters from K-means and PSO-KM is compared to determine which algorithm produces quality clusters. The main parts of this research are: pre-processing, clustering (k-means clustering), PSO Algorithm, hybrid PSO-K-means (PSO-KM) algorithm and Recommendation Algorithm. These parts are discussed in detail below by using diagram:

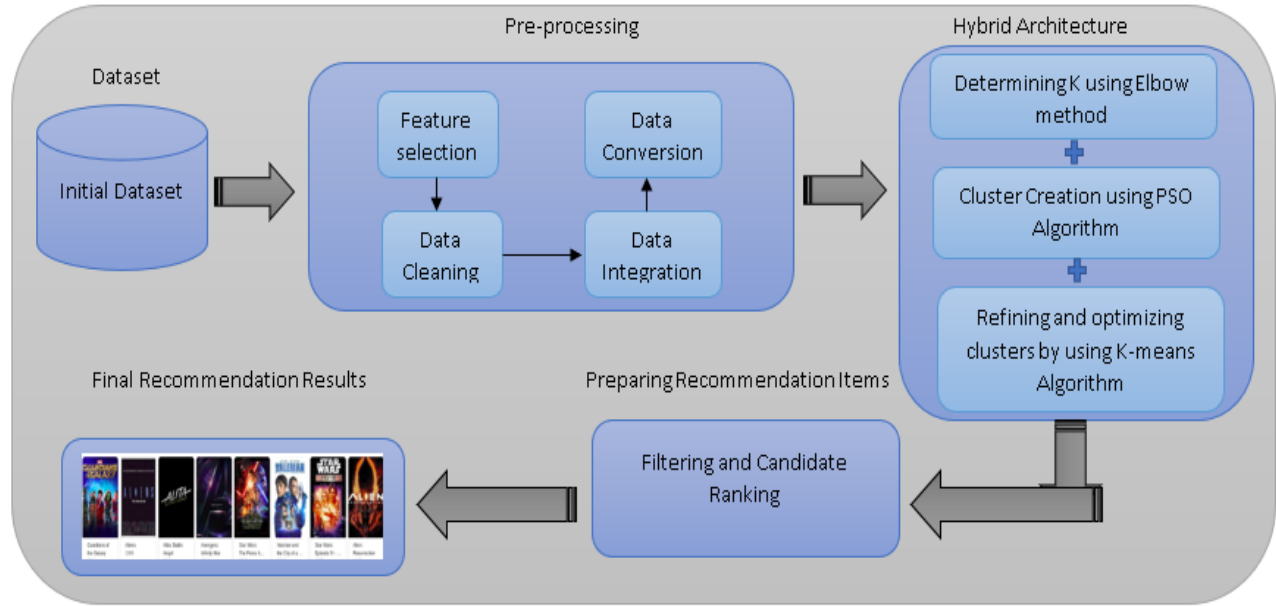


Figure 4.1 Proposed Hybrid Architecture

As it is indicated in the above diagram, before the MovieLens (ml-latest-small) dataset is trained on the Hybrid algorithms, the dataset must pass through the successive stages of Pre-processing. After the dataset is pre-processed, quality and numerical dataset is prepared, now the dataset is ready to be trained on Hybrid algorithms. First, the Elbow method determines the number of clusters or K. Second, the PSO algorithm searches all space for a global solution. When the PSO algorithm finishes by finding the region of global optimum solution, then k-means algorithms start clustering to refine clusters to finalize the clustering task. Finally, after the final clusters are created by Hybrid PSO-KM algorithm, the Popularity-based algorithm by filtering and ranking candidate movies recommends the top 20 or 30 movies for a given user in a given cluster.

4.1.1. Pre-processing

Data pre-processing is an important step of data mining and machine learning techniques that involves transforming raw data (which can be often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors) into an understandable format [38]–[40]. For better accuracy in recommendation results, data should have to pass through different steps of data pre-processing to clean data from inconsistent and missed values and to prepare the data in proper format for the filtering algorithm. The following are the steps the MovieLens (ml-latest-small) dataset should have to pass:

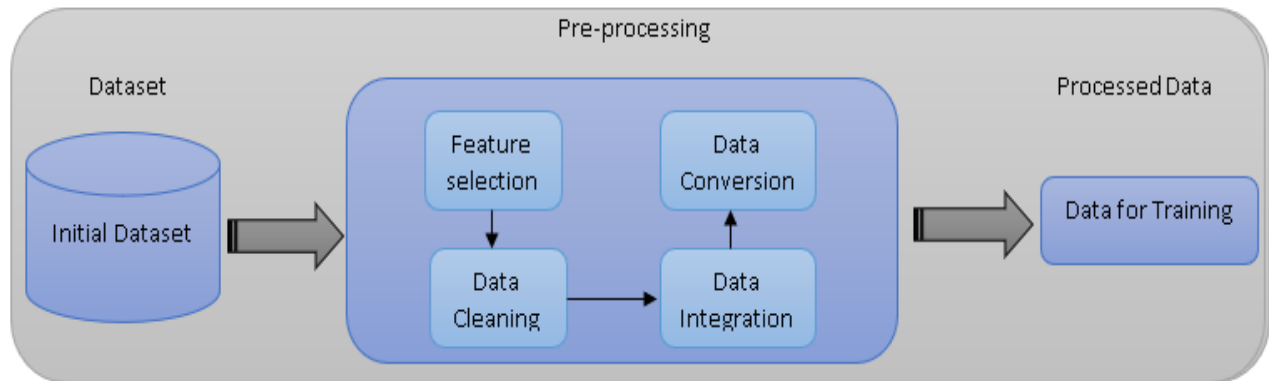


Figure 4.2 Pre-processing steps

4.1.1.1. Feature Selection

Feature selection is a process in which the researcher automatically or manually selects those features which contribute most to the research prediction or recommendation variable or output in which the research is interested in. Using a feature selection process before training data can reduce overfitting, reduce training time and improve accuracy [67]. MovieLens (ml-latest-small) contains the following attributes: user_id, movie_id, movie name, genres, rating timestamp, tags, user rated movies, user rated movies timestamps and user tagged movies timestamps. From all these features based on the time, budget and processing power constraints user_id, movie_id, genres and user rated movies or rating features are used to train the model. These features are selected manually by considering mainly their association to the problem this research tries to solve and their influence on the final recommendation results.

4.1.1.2. Data Integration

Data with different representations are put together and conflicts within the data are resolved. And also, data preprocessing technique can be described as a technique that combines data from multiple sources and provides users a unified representation of these data. In this dataset around four independent excel files exist, these are: links, movies, ratings and tags. From all these independent files movies and ratings are selected. For further preprocessing these independent files should have to be merged by using python merge-tables functionality. Before merge functionality applied these tables, their columns are:

movies: MovieID, Title, Geners

rating: UserID, MovieID, Rating, Timestamp

Since the selected attributes for training the algorithm are user_id, movie_id, genres and user rated movies or rating; Timestamp column from rating table is unwanted and is deleted. After all unwanted columns in each table are deleted, Python merge-tables method t merges the two tables as one table.

4.1.1.3. Data Cleaning

Data is cleansed through processes such as filling in missing values, smoothing the noisy data, or resolving the inconsistencies in the data. Raw data in general may have missing attribute values, containing errors or outliers and may have inconsistent data values. In case of missing data (which is when some data is missing in the dataset) there are few ways to handle these: Ignore the tuple and fill the missing values [74]. When the data is too large and there are too many missing values in the dataset, using ignore the tuple choice is better. In case of numerical datasets and if the data is not that much large, using Filling the missing values (which is by filling the missing values manually, by calculating attribute mean or the most probable value) is a better choice. In MovieLens (ml-latest-small) dataset the type of this dataset is numerical and this dataset is used in recommendation system that requires as many quality information as possible about the customer, the item and their relationship, too many missing values may hinder the recommendation accuracy of the model. And k-means clustering performance is affected by missing values. Because of the K-means missing values performance issue, largeness of the dataset and the need of consistent data, ignoring the tuple choice will be used to resolve the missing values. By using python functionality of selecting all tuples that have missing values all the tuples that have missing values will be deleted. After resolving the missing values problem, the final number of tuples that are left are 1,000,209 numbers of tuples.

4.1.1.4. Data Conversion

Before the k-means clustering algorithm starts, the Genre column should have to be converted to numerical, because the k-means algorithm only uses numerical data to calculate distance between items or movies. To change the Genre column to numerical one way is to convert all Genre values as columns and all the movies that have that Genre type is valued as 1, the rest valued as 0. The following figure shows the conversion of Genre column values to numerical value columns, initially the values of converted Genres are initialized as 0: After the Genres are created as columns, each column is initialized as 1 which are present in the Genres. Finally, the original

categorical Genre column and Title column must be deleted, because K-means clustering only use numerical data to distance calculation

4.1.2. K-means Clustering Algorithm

As it is described in section 2.7.1. [k-mean clustering algorithm](#) is one of the most efficient and most widely used algorithms for solving different problems in different topics because of its simplicity and low complexity [16]. It is an iterative algorithm that tries to create homogenous clusters by representing each cluster by mean or weighted mean known as centroid [46]. The main aim of k-means clustering algorithm is to create clusters that contain homogenous items by using unsupervised data. The basic process of k-means clustering shown below:

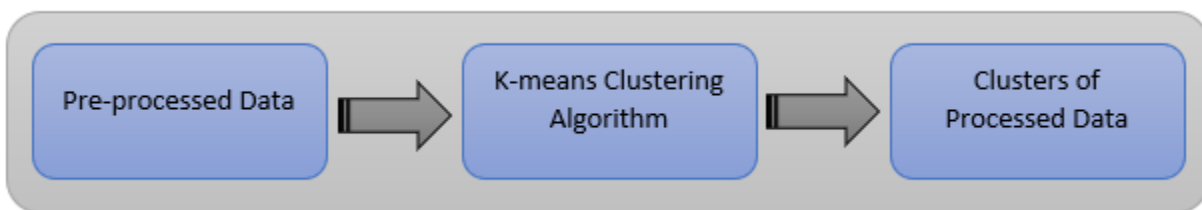


Figure 4.3 k-means clustering algorithm basic process

First, the user chooses **k** centroid points to start with as reference points and then the items are clustered into **k** points based on the closeness of the item to cluster centroid point [46].

Determining the value of K:

Determining the optimal number of clusters in a data set is a fundamental issue in partitional clustering, such as k-means clustering, which requires initial selection of cluster centers to be known and fixed previously before clustering[75], [76]. K-Means is relatively an efficient method, but the final results of clusters are sensitive to initialization and often terminates at a local optimum [73]. Even if there are different methods that are used to give a better result of k (number of clusters) value based on the data used, there is no global method that guarantees to give an exact value of k that can produce an optimized number of clusters. Unfortunately, there is no global theoretical method to find the exact K value or optimal number of clusters, but there are different methods and techniques that try to find the K value that creates the optimal number of clusters. the value of K can be obtained by accurately estimating by using the following techniques. In general, a large k probably decreases the error but increases the risk of overfitting.

The Elbow Method

One of the metrics that is commonly used as a practical method to compare results across different values of K is the mean distance between data points and their cluster centroid. Since increasing the number of clusters will always reduce the distance to data points, increasing K will *always* decrease this metric, to the extreme of reaching zero when K is the same as the number of data points. Thus, this metric cannot be used as the sole target. Instead, mean distance to the centroid as a function of K is plotted and the "elbow point," where the rate of decrease sharply shifts, can be used to roughly determine K . A number of other techniques exist for validating K , including cross-validation, information criteria, the information theoretic jump method, the silhouette method, and the G-means algorithm.

Clustering

To create clusters that contain homogenous items this algorithm uses Euclidean distance calculation to find the correct centroids and reassign items in their closest clusters. The Euclidean distance calculation for finding new centroid points and the reassignment of items to the closest cluster is performed iteratively until all elements are in their closest cluster [44], [46]. The objective function used in k-means aims to minimize total intra-cluster distance [48], [74], [75]:

$$J = \sum_{j=1}^k \sum_{i=1}^n \|x_i^{(j)} - c_j\|^2$$

Equation 4.1 The Computation of objective function

The objective function where $\|X_i^{(j)} - C_j\|$ is a chosen distance measure between a data point $X_i^{(j)}$ and the cluster center C_j , is an indicator of the distance of the n data points from their respective cluster centers or centroids.

By assuming the inputs, i.e. $X_1, X_2, X_3, \dots, X_n$ and value of K , here below are the basic operations that will be done under k-means clustering [14]:

Step one:

First, k (cluster centers or centroids) value will be selected randomly or by using statistical methods such as elbow method or silhouette method. By assuming the centroids $C_1, C_2, \dots, C_k$ as: and C is the set of all centroids.

$$C = C_1, C_2, \dots, C_k$$

Step Two:

By calculating the Euclidean Distance between the point and each centroid from step one, each item is assigned to the closest center. $\text{dist}()$ represent the Euclidean distance.

$$\arg \min_{c_i \in C} \text{dist}(c_i, x)^2$$

Equation 4.2 The Computation of Euclidean Distance

Step Three:

By taking the average of all the points assigned to that cluster the new centroid will be found. S_i is the set of all points assigned to the i^{th} cluster.

$$C_i = \frac{1}{|S_i|} \sum_{x_i \in S_i} x_i$$

Equation 4.3 The Computation of new centroid

Step Four:

In this step, step two and three will be repeated until the clusters remain stable or until none of the cluster assignments change.

Or it can be described as:

For the purpose of this paper, the following symbols are defined:

- N_d : denotes the input dimension, i.e. the number of parameters of each item
- N_o : denotes the number of items to be clustered
- N_c : denotes the number of cluster centroids
- Z_p : denotes the p -th item
- m_j : denotes the centroid of cluster j
- n_j : is the number of data vectors in cluster j
- C_j : is the subset of items that form a cluster.

Using the above symbols, the standard K-means algorithm can be described as:

1. Randomly initialize the cluster centroid items

2. Repeat

- a. For each item, assign the item to the class with the closest centroid item, where the distance to the centroid is determined using:

$$d(Z_p, m_j) = \sqrt{\sum_{k=1}^{N_d} (Z_{pk} - m_{jk})^2} \quad \dots\dots\dots (1)$$

Equation 4.4 Equation 4.4. the Computation of Euclidean Distance

where k subscripts the dimension

- b. Recalculate the cluster centroid items, using:

$$m_j = \frac{1}{n_j} \sum_{\forall Z_p \in C_j} Z_p \quad \dots\dots\dots (2)$$

Equation 4.5 The Computation of new centroid

until a stopping criterion is satisfied, i.e. until the clusters remain stable or until none of the cluster assignments change.

K-means Convergence

The main steps k-means algorithm use to cluster are [41], [75]:

1. Place K points into the space represented by the items that are being clustered. These points represent initial group centroids.
2. Assign each item to the group that has the closest centroid.
3. When all objects have been assigned, recalculate the positions of the K centroids.
4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the items into groups from which the metric to be minimized can be calculated.

The flow chart of k-means algorithm that shows the order of basic operations of the clustering process is shown below [76]:

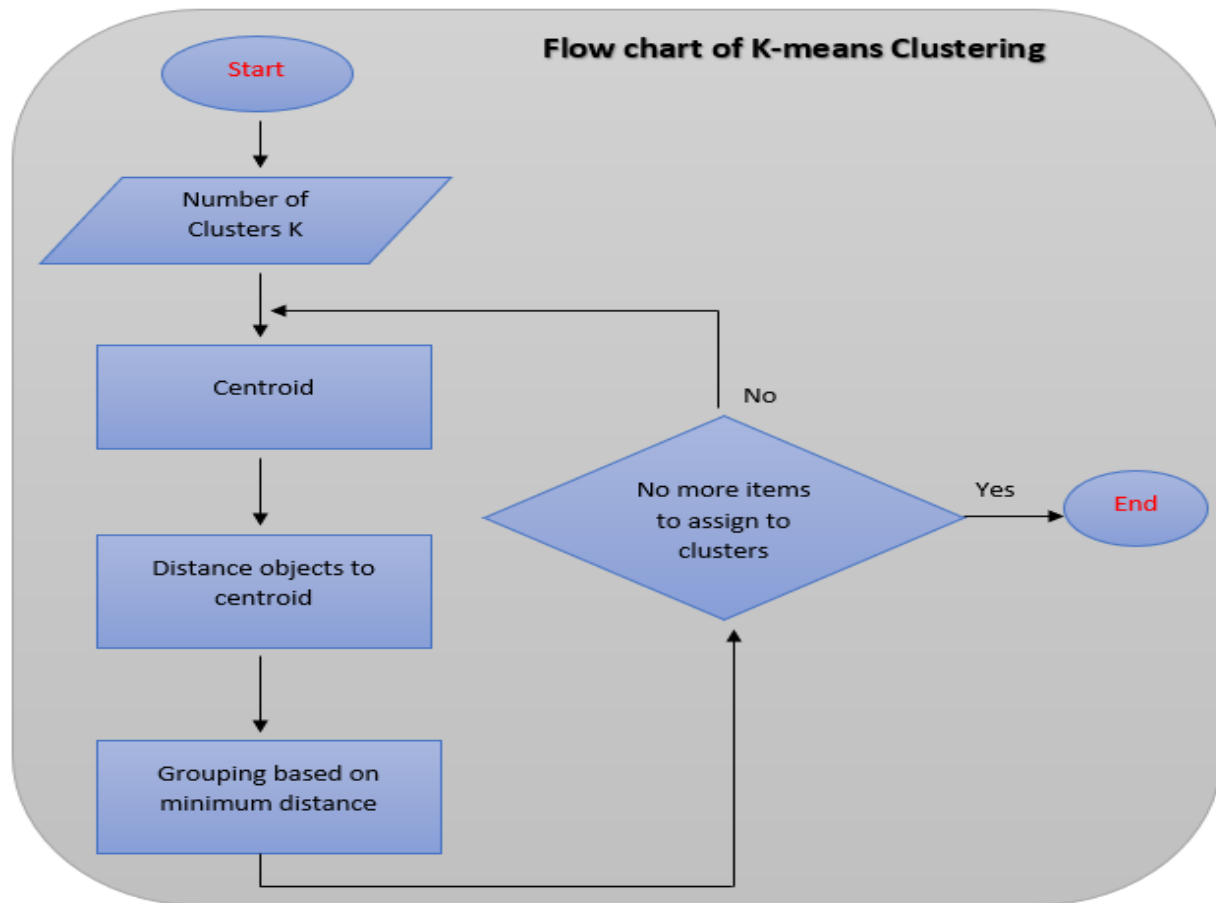


Figure 4.4 Flow chart of k-means clustering algorithm basic process.

4.1.3. Particle Swarm Optimization Algorithm (PSO)

As described in section 2.8. **PSO is one** of evolutionary and population-based search algorithms which simulates the movement and flocking of birds [15], [59], [60]. The birds in a flock are symbolically represented as particles, which is initialized with a population of random solutions. The algorithm maintains a population of particles which seems as simple agents moving through a problem space, where each particle represents a potential solution to an optimization problem [11], [12]. A particle's location in the multi-dimensional problem or function search space represents one candidate solution for the problem. Each particle has memory and fly with a certain velocity and also moves in problem space to find the global best position in an iterative process. At each iteration of the algorithm, the velocity vector for each particle is modified based on three

conditions: the particle momentum, the best position reached by the particle and that of all particles up to the current stage [59]. In the next iteration, the particle is moved to its next position based on the determined velocity for each particle. When a particle moves to a new position or location, a different solution for the problem is generated. Then this solution is evaluated by a fitness function to provide a quantitative value of the solution's utility. The PSO algorithm consists of three general steps, these are: generation of particles (generating particle's positions) and their information (velocities), movements (velocity update) and new information vector (position update) [15].

Each particle represents a position in N_d dimensional problem space, and is flown through this multi-dimensional problem search space, adjusting its position toward both [11]:

- the particle's best position found so far and
- the best position in the neighborhood of that particle

Each particle i maintains the following information:

- ◆ x_i : The current position of the particle;
- ◆ v_i : The current velocity of the particle;
- ◆ y_i : The personal best position of the particle.

Using the above symbols, the particle's best position is adjusted according to the following formula:

$$v_{i,k}(t+1) = wv_{i,k}(t) + c_1r_{1,k}(t)(y_{i,k}(t) - x_{i,k}(t)) + c_2r_{2,k}(t)(y_k(t) - x_{i,k}(t)) \dots\dots 3$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \dots\dots 4$$

Equation 4.6 The Computation of particle's best position

where w is the inertia (within cluster sum of squares criterion) weight, c_1 and c_2 are the acceleration constants, $r_{1,j}(t), r_{2,j}(t) \sim U(0,1)$ and $k = 1, \dots, N_d$.

The velocity is thus calculated based on the following three contributions: (1) a fraction of the previous position velocity, (2) the cognitive component which is a function of the distance of the particle position from its personal best previous position, and (3) the social component which is

a function of the distance of the particle position from the best particle position found so far (i.e. the best of the personal bests).

Equation (3) shows the gbest version of PSO, where for each particle in the swarm the neighborhood is simply the entire swarm or the entire items. Then the social component causes particles to be drawn toward the best particle in the swarm.

The personal best position of particle i is calculated as:

$$y_i(t+1) = \begin{cases} y_i(t) & \text{if } f(x_i(t+1)) \geq f(y_i(t)) \\ x_i(t+1) & \text{if } f(x_i(t+1)) < f(y_i(t)) \end{cases} \quad \text{.....5}$$

Equation 4.7 The Computation of particle's personal best position

The PSO usually executes equations (3) and (4) until a specified number of iterations has been exceeded or it is terminated when the velocity updates are close to zero over a series number of iterations.

In clustering, a single particle represents items contained in a given cluster or related by one N_c cluster centroid. i.e., each particle x , is constructed as follows:

$$x_i = (m_{i1}, \dots, m_{ij}, \dots, m_{iN_c}) \quad \text{.....(6)}$$

where m_{ij} refers to the j^{th} cluster centroid of the i^{th} particle in cluster C_{ij} . Therefore, a swarm represents a number of candidate clusters for the current items.

To measure the fitness of particles, quantization error can be used as follows:

$$J_e = \frac{\sum_{j=1}^{N_c} [\sum_{p \in C_{ij}} d(z_p, m_j)] / |C_{ij}|}{N_c} \quad \text{..... (7)}$$

Equation 4.8 The Computation of the fitness of particles

where d is defined in equation (1) above, and $|C_{ij}|$ is the number of items belonging to cluster C_{ij} , i.e. the frequency of that cluster.

The PSO algorithm can be summarized as [12]:

- (1) At the initial stage, each particle randomly chooses k different items from the item collection as the initial cluster centroid vectors.
- (2) For each particle:
 - (a) Assign each item in the item set to the closest centroid vector.
 - (b) Calculate the fitness value based on equation (7).
 - (c) Using the velocity and particle position to update equations (3) and (4) and to generate the next solutions.
- (3) Repeat step (2) until one of the following termination conditions is satisfied:
 - (a) The maximum number of iterations is exceeded or
 - (b) The average change in centroid vectors is less than a predefined value.

The flow chart of PSO algorithm displayed below shows the order of basic operations of the PSO algorithm:

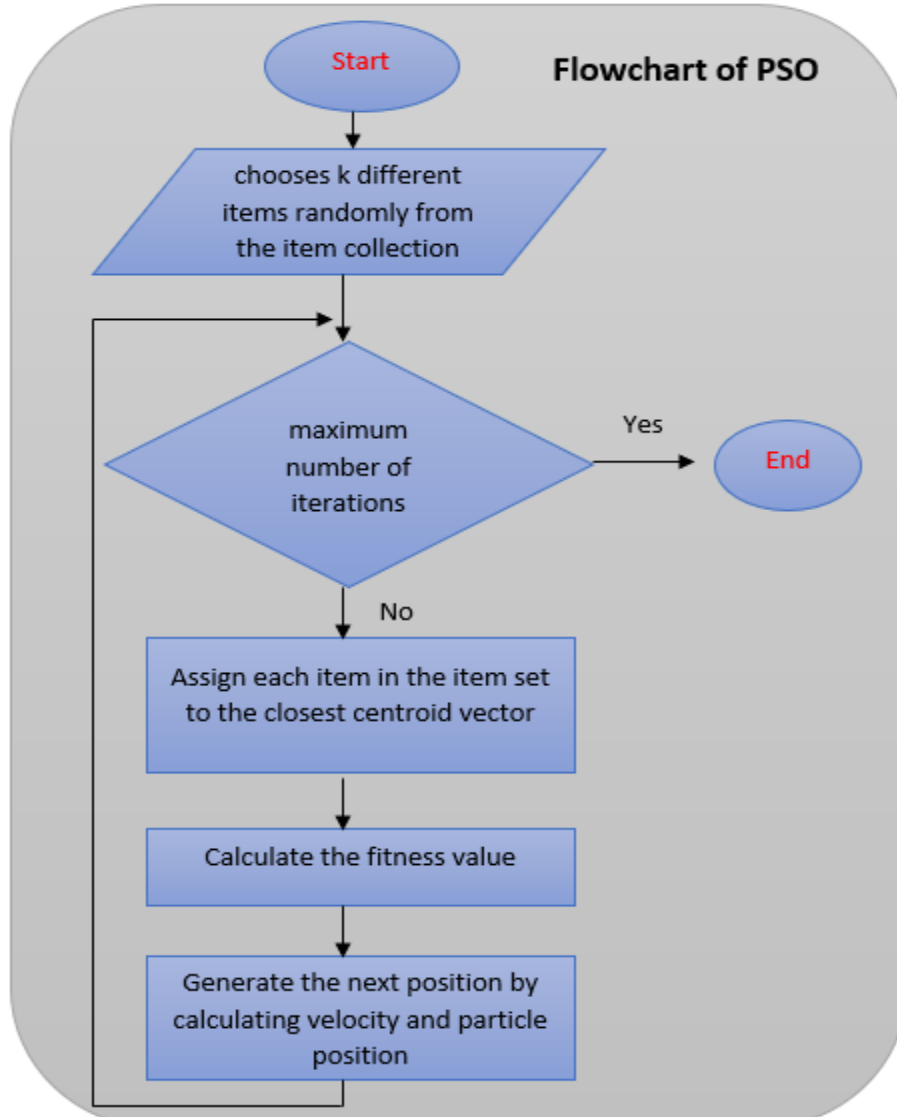


Figure 4.5 Flow chart of PSO algorithm basic process

4.1.4. Hybrid Particle Swarm Optimization Algorithm and K-means

The limitation of using k-means clustering algorithm is: it depends on the initial selection of K number of clusters and the initial cluster centroid, because of its sensitivity to initial centroid or partition it only can generate local optimal solution [53]. The limitation of the PSO algorithm is: it converges slowly when the search space is high or the data is large, it converges in mid optimum points prematurely and it doesn't perform well when the data is too large or complex [15]. To improve the limitations of both algorithms and to use the advantage of both algorithms a hybrid PSO-K-means algorithm is proposed in this research paper.

The basic process of hybrid PSO-K-means algorithm shown below:

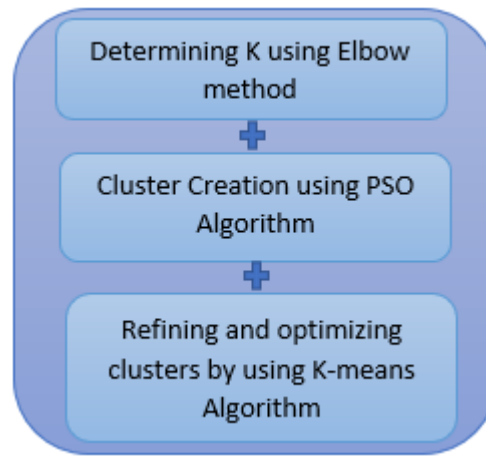


Figure 4.6 Hybrid PSO-K-means algorithm steps

As most nature-based optimization algorithms, PSO is a global search algorithm, which has a strong ability to find global optimistic results. But, the PSO algorithm converges very slowly near to the solution. On the other hand, K-means Algorithm converges fast to a local optimum result, but its ability to find the global solution is weak [59]. Due to its iterative nature to find the global optimum solution, it converges slowly and also can generate more compact clustering results from the low dimensional dataset than the traditional K-means clustering algorithm.

By combining these two algorithms, i.e. PSO-K-means (PSO-KM), first by using the Elbow Method the number of cluster K is determined, then the PSO algorithm searches all space for a global solution. When the PSO algorithm finishes by finding the region of global optimum solution, then k-means algorithms start clustering to refine clusters to finalize the clustering task. The ability of PSO algorithms to find global optimum solutions can be used in K-Mean algorithms to find the local optimal solution [15]. Even if the PSO algorithm generates quality clusters than the K-means does, in terms of execution time, the K-means algorithm is more efficient for large datasets or complex data [12]. This hybrid algorithm accelerates the convergence speed as well as accuracy by using the PSO algorithm to discover the vicinity of the optimal solution by a global search and the result of PSO is used as the initial seed of the K-means algorithm. Finally, K-means algorithms refine clusters and generate the final result.

The hybrid PSO-K-means (PSO-KM) algorithm can be summarized as:

- (1) Start the PSO clustering process to find global optimum solution until the maximum number of iterations is exceeded
- (2) Inherit clustering results from PSO as the initial centroid vectors of K-means algorithm.
- (3) Start K-means process to find local optimum solution until maximum number of iterations is reached.

4.1.5. Recommendation Algorithm

As it is described in 2.9. Recommendation systems are helper systems that allow systems to take advantage of the enormous amount of information that is available in the internet and to filter and prioritize items to recommend to users based on their interest. Recommendation systems are one of Artificial Intelligence based algorithms that search through all possible options and create a personalized list of items that are interesting and relevant to an individual. These results are based on their profile, what other users with similar traits/demographics are watching, search/browsing history and how likely are the users to watch those movies. The information overload is not a problem only in case of research, it is also a problem in the entertainment business. Nowadays recommendation systems are developed to recommend users about web pages, movies, music, restaurants, news articles and tv programs by understanding users' interest. Recommending movies to users based on their interest is very difficult because of the nature of details or features of the movies, such as: country, genre, year, participants (actor, directors, script writers) and movie company.

Popularity-based Recommendation Algorithm

From all the types of Recommendation algorithms Popularity-based Recommendation Algorithm type is one mostly used Recommendation Algorithm in top review websites like Rotten Tomatoes and IMDb to maintain a database of movies and their popularity in terms of ratings and reviews. The major merit of this popularity-based recommendation system is that it eliminates the need for information like user browsing history, user preferences, genre, the star cast of the movie, and other factors. The recommendation algorithm steps proposed in this paper shown as follows:

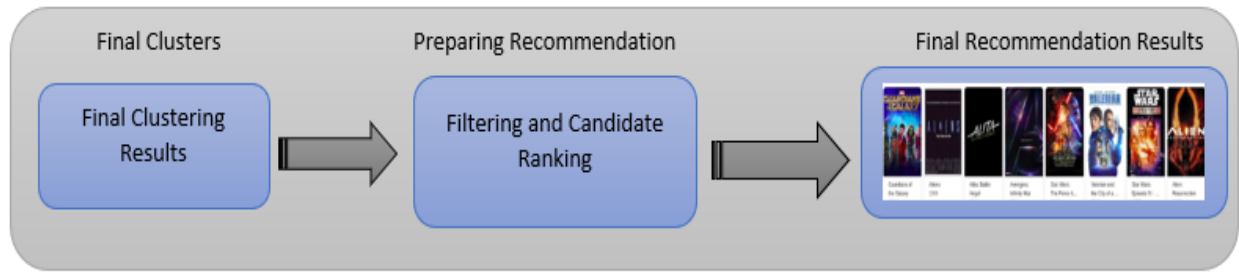


Figure 4.7 Recommendation Algorithm Steps

After the hybrid algorithm i.e. PSO and K-means algorithm finished creating clusters, then the recommendation algorithm starts by accepting given user's ID and cluster, to recommend movies that the given user didn't rate before and that are popular in that cluster. First, for all movies in the data average ratings of each movie is calculated and this is used in selecting popular movies in the given cluster. Second, all the movies the given user didn't see in the given cluster is collected. From this collected list the popular movies in the given cluster is sorted in descending order by using average movie rating calculated earlier. Finally, the algorithm recommends to the given user the top 20 or 30 movies in the top of sorted popular movies.

The Popularity-based recommendation algorithm can be summarized as [58][57]:

- (1) Calculate average movie ratings for each movie
- (2) Collect all the movies the given user didn't rate before
- (3) Sort the list in step (2) by using average movie rating calculated in step (1)
- (4) Recommend the top 20 or 30 movies to the given user from sorted list in step (3)

Chapter Five

5. Implementation of the Proposed Model

This chapter presents the implementation of the proposed solution of Hybrid PSO-K-means (PSO-KM) by using the methodology discussed in chapter three and also, implement the proposed solution in chapter four. This chapter shows the implementation and comparisons of K-means clustering algorithm and hybrid PSO-K-means (PSO-KM) algorithm to determine which algorithm Produces quality clusters. First, it shows the implementation of the pre-processing stage which tries to prepare the dataset for missing and inappropriate values and produces quality dataset for clustering by passing dataset through different pre-processing stages. Second, before the K-means clustering or Hybrid PSO-K-means (PSO-KM) algorithm starts, it shows the implementation of the Elbow Method which is used to determine the number of clusters or K for the dataset. Third, it shows the implementation of K-means clustering algorithm which produces homogenous clusters. Fourth, it shows the implementation of Hybrid PSO-K-means (PSO-KM) algorithm which implements both PSO and K-means algorithm as hybrid nature to produce quality clusters. Fifth, it shows the implementation of evaluation mechanisms such as Silhouette Score, the Sum-of Squares-Error (SSE) criterion and Quantization Error to determine which method (i.e. K-means or Hybrid PSO-KM) produce quality clusters. Sixth, it finally shows the implementation of Popularity-based recommendation algorithm which recommends movies to given user which the user didn't see before based on movies popularity in the cluster the given user belongs to. The dataset used to train is the famous MovieLens (ml-latest-small) dataset.

5.1. Implementation Environment

This research paper used several machine learning tools and packages to implement the proposed solution i.e. Hybrid PSO-KM. This paper uses python programming language for implementing from the data preprocessing to the model building phase and also for evaluation phase for evaluating both K-means and Hybrid PSO-KM algorithms. Because of python's favorability from the developer side in all web development applications to scripting and process animation and in Artificial intelligence, machine learning, and deep learning areas is one of the reasons python is used in this research paper. The diverse application of the Python language, the availability of enormous its own- and third-party packages, and also its scalability and user-friendly nature are few reasons why python is preferred in machine learning and artificial language applications.

Table 5.1 shows the list of tools and python packages with their version and description which is used in this research paper.

Table 5.1 Description the Tools and Python Packages Used During the Implementation

No.	Tools Name	Version	Description
1	Anaconda Navigator	1.9.12	allows us to launch development applications and easily manage conda packages, environments, and channels without the need to use command-line commands.
2	Jupyter notebooks	6.0.3	An open-source web application that allows us to create and share documents that contain live code, equations, visualizations, and narrative text. Uses include data cleaning and transformation, numerical simulation, statistical modeling, data visualization, and machine learning.
3	Python	3.8	User-friendly, easy to learn and powerful programming language to develop a machine learning application.
4	Scikit-learn	0.23	A set of python modules for machine learning and data mining. This study uses it for feature extraction and training and testing models. The name of the package is called sklearn.
5	pandas	1.0.3	High-performance, easy-to-use data structures, and data analysis tools. This study uses it for data reading, manipulation, writing and handling the data frame.
6	NumPy	1.18.1	Array processing for numbers, strings, and objects. This study uses it for handling converting the text to numeric data for features and training and testing the model.

7	matplotlib	3.1.3	Publication quality figures in python. This study uses it for data and results visualization.
8	utils	1.0.1	A collection of small Python functions and classes which make common patterns shorter and easier.
9	Microsoft Excel	2019	Used data preparation tasks pre-processing stages and also to store the final results.

5.2. Deployment Environment

The tools which are discussed above in section 5.1 have been deployed on a personal computer equipped with a processor Intel(R) Core (TM) i7-8550U, CPU 1.80GHz, 2 Core(s), 8 Gigabyte of physical memory, 1 Terabyte hard disk storage capacities. The operating system is Windows 10 Pro, 64 bits.

5.3. Dataset Description

The dataset used for clustering is the well-known MovieLens (ml-latest-small) dataset for its quality and used on so many researches. The dataset is prepared by GroupLens Research Group in University of Minnesota on September 26, 2018 [13]. This MovieLens (ml-latest-small) dataset is a smaller version of GroupLens Research Group prepared datasets. This dataset (ml-latest-small) describes 5-star rating and free-text tagging activity from [MovieLens](http://movielens.org), a movie recommendation service. It contains 100836 ratings and 3683 tag applications across 9742 movies. These data were created by 610 users between March 29, 1996 and September 24, 2018. This dataset was generated on September 26, 2018. Users were selected at random for inclusion. All selected users had rated at least 20 movies. No demographic information is included. Each user is represented by an id, and no other information is provided. The data are contained in the files ‘links.csv’, ‘movies.csv’, ‘ratings.csv’ and ‘tags.csv’.

As discussed in section 4.1.1. Due to time, budget, processing power and memory space constraints ‘movies.csv’, ‘ratings.csv’ files are selected to train the model. From these files’ user_id, movie_id, genres and user rated movies or rating features are used to train the model. Other features from these files are deleted by using python drop-column functionality. The two

independent files will be merged by python merge-tables functionality. After merging the number of tuples is 1,000,386.

Before the training the model process starts, all missing value rows or tuples should have to be cleaned. After all missing value rows are cleaned the number of tuples is 1,000,209. After the pre-processing stage is completed, the remaining columns are:

- MovieID -: unique identification number for each movie
- UserID -: unique identification number for each user
- Title -: title for each movie
- Geners -: movie type for each movie (a movie can have more than one genre)
- Rating -: users rank (1-5) given for each movie

5.4. Pre-processing Implementation

To implement the proposed model, first the data must pass through pre-processing stages for creating quality data. In pre-processing stages python programming language packages and modules are used. To implement the pre-processing and the model, the following activities should be completed. The following figure i.e. figure5.1 shows the implementations of importing library packages and loading the dataset. The loaded dataset using panda is used throughout the whole implementation of this study.

```
#Loading the datasets
import pandas as pd
data1 = pd.read_csv("ml-latest-small/ml small movies data/movies data.csv")
data2 = pd.read_csv("ml-latest-small/ml small movies data/ratings data.csv")
```

Figure 5.1 Python Code for Loading the Dataset

5.4.1. Feature Selection Implementation

Before merging the two files, deleting unwanted columns from both files is necessary. The python method used to delete unwanted columns from a selected file is *drop ()*. By giving the columns name from selected file the *drop ()* method deletes the columns as shown below:

```
#Deleting unnecessary columns
data1 = data1.drop('Unnamed: 0',axis = 1)
data2 = data2.drop(['Unnamed: 0','Timestamp'],axis = 1)
```

Figure 5.2 Python Code for Deleting unwanted Columns

5.4.2. Data Integration Implementation

Before training the model, the two independent files should have to be merged as one file. The python *merge ()* method merges the two independent files by accepting the two file names, their common attribute or column name and how the merging should be done. The following picture shows the implementation of *merge ()* method.

```
#Merging or Integrating both the datasets
data = pd.merge(data2 , data1 , how='outer', on='MovieID')
```

Figure 5.3 Python Code for Deleting unwanted Columns

5.4.3. Data Cleaning Implementation

After the two independent files are merged as one, now all the missing value rows must be deleted for the purpose of training the model with quality data. The python's *dropna ()* method drops or deletes all rows that have missing values in the dataset. The following picture shows the implementation of *dropna ()* method.

```
# Deleting all rows with missing values
data.dropna(inplace=True )
```

Figure 5.4 Python Code for Deleting Rows with Missing Values

5.4.4. Data Conversion Implementation

Before the k-means clustering algorithm starts, the Genre column should have to be converted to numerical, because the k-means algorithm only uses numerical data to calculate distance between items or movies. To change the Genre column to numerical one way is to convert all Genre values as columns and all the movies that have that Genre type is valued as 1, the rest valued as 0. The following figure shows the conversion of Genre column values to numerical value columns, initially the values of converted Genres is initialized as 0:

```
# Converting Genres into different columns and initialize values as 0
x = data.Genres
a = list()
for i in x:
    abc = i
    a.append(abc.split('|'))
a = pd.DataFrame(a)
b = a[0].unique()
for i in b:
    data[i] = 0
```

Figure 5.5 Python Code for Converting Genre values to Columns

After the Genres are created as columns, each column is initialized as 1 which are present in the Genres. The following picture shows the initialization of 1 for Genres:

```
# assign 1 to all the columns which are present in the Genres
for i in b:
    data.loc[data['Genres'].str.contains(i), i] = 1
```

Figure 5.6 Python Code for Initializing Genre values

Finally, the original categorical Genre column and Title column must be deleted, because K-means clustering only uses numerical data to distance calculation. The following picture shows implementation of the removal of the two columns:

```
# Deleting Genres and Title columns
data = data.drop(['Genres', 'Title'], axis = 1)
data.head()
```

Figure 5.7 Python Code for Deleting Genres and Title Column

5.5. K-means Clustering Algorithm Implementation

As described in previous chapters, k-means clustering algorithm is one popular clustering algorithm used for grouping or classifying problems. Since k-means clustering result is dependent on the quality of input dataset, the dataset must be pre-processed. After completing the pre-processing stage as shown in section 5.4., now k-means clustering algorithm starts to create homogenous clusters.

5.5.1. Determining K (number of Clusters)

From a number of limitations of k-means algorithm, one is its sensitivity to the number of clusters K and also the user must give the initial K value before K-means starts. K-Means is relatively an

efficient method, but the final results of clusters are sensitive to initialization and often terminates at a local optimum [73]. Unfortunately, there is no global theoretical method to find the exact K value or optimal number of clusters, but there are different methods and techniques that try to find the K value that creates the optimal number of clusters. One of the metrics is the Elbow Method that is used to compare results across different values of K is the mean distance between data points and their cluster centroid. Since increasing the number of clusters will always reduce the distance to data points, increasing K will *always* decrease this metric, to the extreme of reaching zero when K is the same as the number of data points. Thus, this metric cannot be used as the sole target. Instead, mean distance to the centroid as a function of K is plotted and the "elbow point," where the rate of decrease sharply shifts, can be used to roughly determine K . the following figure shows the implementation of Elbow Method to determine K :

```
# Choosing the optimal k
k_range = range(1,20)
x = data
# clustering the data for k values ranging 1 to 20
k_means_var = [KMeans(n_clusters = k).fit(x) for k in k_range]
centroids = [X.cluster_centers_ for X in k_means_var]

k_euclid = [cdist(x, cent, 'euclidean') for cent in centroids]
dist = [np.min(ke, axis=1) for ke in k_euclid]

# Calculate within-cluster sum of squares
wcss = [sum(d**2) for d in dist]
```

Figure 5.8 Python Code for Determining K by using Elbow Method

The following figure visualizes Elbow Points against the number of clusters and this graph is used to determine the number of cluster K by seeing the number of cluster points where the Elbow Points become constant.

```
# Visualize the elbow method for determining k
import matplotlib.pyplot as plt
plt.plot(k_range, wcss, marker='o')
plt.xlabel('Number of clusters')
plt.ylabel('Elbow Score')
plt.title('Elbow Curve')
plt.grid()
```

Figure 5.9 Python Code for visualizing Elbow points against number of clusters

5.5.2. K-means Clustering Algorithm

After all the data pre-processing is completed and the number of Cluster K is determined, the K-mean algorithm is now ready to create clusters. To create clusters that contain homogenous items this algorithm uses Euclidean distance calculation to find the correct centroids and reassign items in their closest clusters. The Euclidean distance calculation for finding new centroid points and the reassignment of items to the closest cluster is performed iteratively until all elements are in their closest cluster[44], [46]. The following figure shows the process of creating clusters and also predicting clusters by using K-means algorithm, number of clusters K used 8:

```
# k-means creating 8 homogenous clusters
kmeans = KMeans(n_clusters = 8)
#to create clusters
kmeans.fit(data_scaled)
#to predict clusters
pred = kmeans.predict(data_scaled)
```

Figure 5.10 Python Code for Creating Clusters by using K-means Algorithm

5.6. Hybrid Particle Swarm Optimization and K-means Clustering Algorithm (PSO-KM) Implementation

To build machine learning models using the proposed algorithm, the following are all the python packages including sci-kit learning library package and other packages imported and used. And also, all the packages that are used for evaluating both k-means and hybrid PSO-KM cluster results. The following figure shows the packages and libraries imported and used for training the model and evaluating the model:

```

#All the header files required for the Model
from pso import ParticleSwarmOptimizedClustering
from particle import quantization_error, calc_sse
from kmeans import KMeans
from sklearn.metrics import silhouette_score
from utils import normalize
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.sparse import csr_matrix
import helper
from factor_analyzer import FactorAnalyzer
from sklearn import metrics
import random

```

Figure 5.11 Python Code for Importing the required header files

By combining these two algorithms, i.e. PSO-K-means (PSO-KM), first by using the Elbow Method the number of cluster K is determined, then the PSO algorithm searches all space for a global solution. When the PSO algorithm finishes by finding the region of global optimum solution, then k-means algorithms start clustering to refine clusters to finalize the clustering task. This hybrid algorithm accelerates the convergence speed as well as accuracy by using the PSO algorithm to discover the vicinity of the optimal solution by a global search and the result of PSO is used as the initial seed of the K-means algorithm. Finally, K-means algorithm refine clusters and generate the final result.

First the PSO algorithm starts to find a global optimum solution that can be used in the K-Mean algorithm. Generally, the PSO algorithm consists of three steps, these are: first, generation of particles (generating particle's positions) and their information (velocities). The PSO maintains a population of particles which seems as a simple agent moving through a problem space, where each particle represents a potential solution to an optimization problem. A particle's location in the multi-dimensional problem or function search space represents one candidate solution for the problem. Second, particle movements (velocity update). Each particle has memory and fly with a certain velocity and also moves in problem space to find the global best position in an iterative process. At each iteration of the algorithm, the velocity vector for each particle is modified based on three conditions: the particle momentum, the best position reached by the particle and that of all particles up to the current stage. Third, particles have a new information vector (position update). In the next iteration, the particle is moved to its next position based on the determined

velocity for each particle. When a particle moves to a new position or location, a different solution for the problem is generated. Then this solution is evaluated by a fitness function to provide a quantitative value of the solution's utility. Finally, the k-means algorithm uses the centroids created by the PSO algorithm as a seed to refine and create the final clusters as indicated in section 5.5.2.

The following figure shows the implementation of the hybrid PSO-KM algorithm. The *ParticleSwarmOptimizedClustering()* method initializes fields or variables that will be used for particle movement to find pbest (personal best) and gbest (global best) coordinates or solutions. These are: number of particles, number of particles, dataset, hybrid, maximum iteration and print debug option. Based on iterative iteration on this dataset, using 10 numbers of particles yields good centroids or global solutions. The *run()* method starts particle movement in the search space to find pbest (personal best) and gbest (global best) coordinates or solutions and calls all the necessary methods that determine to find global optimum solution iteratively . The figure only shows the main methods and other methods will be included in Appendices part of this document.

```
#Hybrid PSO-KM implementation no particle=10
pso_rep = ParticleSwarmOptimizedClustering(
    n_cluster=8, n_particles=10, data=data_norm, hybrid=True,
    max_iter=2000, print_debug=2000)
pso_rep.run()
pso_kmeans = KMeans(n_cluster=8, init_pp=False)
pso_kmeans.centroid = psorep.gbest_centroids.copy()
predicted_pso_rep = psorep.predict(data_norm)
```

Figure 5.12 Python Code for Implementing PSO-KM algorithm

5.7. Model Evaluation Implementation

Clustering algorithms are more difficult to evaluate. Internal metrics [1] use only information on the computed clusters to evaluate if clusters are compact and well-separated. Also, you can have external metrics that perform a statistical testing on the structure of your data. In this study to evaluate the quality of cluster, the methods used are: intra-cluster distance (Sum-of Squares-Error (SSE)) to determine the distance of items in cluster from the centroid, inter-cluster distance (Silhouette score) to determine the distance of neighboring clusters from each other and fitness of algorithm (Quantization Error) to determine which algorithm minimizes the average (squared) quantization error.

The range of silhouette score (inter-cluster distance) is from -1 to 1. The better result of silhouette score is 1, which depicts the neighboring clusters are farthest to each other. The following figure shows the implementation of silhouette score:

```
# Calculating silhouette_score
print('Silhouette:', silhouette_score(data_norm, predicted_kmeans))
```

Figure 5.13 Python Code for Implementing Silhouette Score.

Sum-of Squares-Error (SSE) determines from two algorithms which algorithm produces compact clusters. The smallest the distance the better the cluster quality is. The following figure shows the implementation of Sum-of Squares-Error (SSE):

```
# Calculating Sum-of Squares-Error (SSE)
def calc_sse(centroids: numpy.ndarray, labels: numpy.ndarray, data: numpy.ndarray):
    distances = 0
    for i, c in enumerate(centroids):
        idx = numpy.where(labels == i)
        dist = numpy.sum((data[idx] - c)**2)
        distances += dist
    return distances
```

Figure 5.14 Python Code for Implementing Sum-of Squares-Error (SSE)

The fitness of algorithm (Quantization Error) determines from two algorithms which algorithm has good fitness to produce quality clusters. The smaller the result the better fitness the algorithm has. The following figure shows the implementation of Quantization Error:

```
def quantization_error(centroids: np.ndarray, labels: np.ndarray, data: np.ndarray) -> float:
    error = 0.0
    for i, c in enumerate(centroids):
        idx = np.where(labels == i)
        dist = np.linalg.norm(data[idx] - c)
        dist /= len(idx)
        error += dist
    error /= len(centroids)
    return error
```

Figure 5.15 Python Code for Implementing Quantization Error

5.8. Popularity-based Recommendation Algorithm Implementation

Recommendation systems are one of Artificial Intelligence based algorithms that search through all possible options and create a personalized list of items that are interesting and relevant to an individual. These results are based on their profile, what other users with similar

traits/demographics are watching, search/browsing history and how likely are the users to watch those movies. From all the types of Recommendation algorithms Popularity-based Recommendation Algorithm type is one mostly used Recommendation Algorithm. By accepting the user_ID and cluster number this algorithm computes the following: First, for all movies in the dataset average ratings of each movie is calculated and this is used in selecting popular movies in the given cluster. Second, all the movies the given user didn't see in the given cluster is collected. From this collected list the popular movies in the given cluster is sorted in descending order by using average movie rating calculated earlier. Finally, the algorithm recommends to the given user the top 20 or 30 movies in the top of sorted popular movies.

First, before going to filtering the dataset, the Title attribute or columns is required and it needs to be merged with the dataset (with which clusters are formed) i.e. data dataset. So, the first independent dataset before merging i.e. data1 dataset is used to merge with the dataset (with which clusters are formed) i.e. data dataset. Also, the Genres attribute or column is deleted and then all the missing value rows are deleted. The following figure shows the implementation of the above concept:

```
data3=data1
data3 = data3.drop(['Genres'],axis =1)
data3.dropna(inplace= True )
recommend1=pd.merge(data,data3,how='outer',on='MovieID')
recommend1.isnull().sum()
recommend1.dropna(inplace= True )
```

Figure 5.16 Python Code for Implementing merging data1 with data dataset

Second, the independent dataset i.e. data2 dataset is used to calculate average movie ratings each movie and this is used to filter movies based on their popularity. After average movie ratings are calculated it is merged with the above recommend1 dataset. The following figure shows the implementation of calculating average movie ratings and merging with recommend1 dataset:

```
movie_rating = data2.groupby(['MovieID'], as_index=False)
average_movie_ratings = movie_rating.agg({'Rating':'mean'})
average_movie_ratings.columns=['MovieID', 'AvgMovieRating']
recommend2=pd.merge(recommend1,average_movie_ratings,how='outer',on='MovieID')
```

Figure 5.17 Python Code for Implementing merging data2 with recommend1 dataset

Finally, now the filtering and candidate ranking algorithm starts to prepare movie recommendation results. First, any user from a given cluster is selected and the algorithm tries to filter all the movies the user rate in the given cluster. By using those movies, the algorithm filters all the movies the user didn't rate or saw before in that cluster. Finally, the algorithm ranks those movies based on their popularity by using average movie ratings calculated earlier and now the algorithm selects the top 20 movies from the ranked lists and recommends them for the given user. The following figure shows the implementation of filtering and ranking algorithm and for this implementation the following data is used i.e. UserID=10, cluster=2:

```
UserID=10.0
cluster=2
#Identifying movies which have been rated by the current user
user_mv=recommend2.iloc[list(np.where(User_label==2))[0]]
#Identifying movies which have not been rated by the current user
user_new_movies=user_mv[user_mv['UserID']!=UserID]
#selecting the unique movies from user_new_movies
unique_movies= user_new_movies.drop_duplicates('MovieID')
#creating dataframe for displaying the list of movies
movielist = pd.DataFrame(unique_movies, columns = ['Title','AvgMovieRating'])
movielist.sort_values('AvgMovieRating', ascending=False, inplace=True)
movielist.set_index('AvgMovieRating', inplace=True)
movielist.head(20)
```

Figure 5.18 Python Code for Implementing for filtering and ranking movie lists

Chapter Six

6. Result and Discussion

In this chapter, the results of the experiments of the proposed solution for K-means and Hybrid PSO-KM algorithm will be discussed. The result of the Elbow method for finding appropriate K (number of clusters) will be discussed. The result of the data pre-processing process and the result of K-mean clustering and Hybrid PSO-KM algorithm will also be compared and discussed. Finally, the result of Popularity-based Recommendation will be discussed.

6.1. Data Pre-processing Result

6.1.1. Feature Selection Stage Result

To implement the proposed model, first the data must pass through pre-processing stages for creating quality data. First, after all the dataset are loaded (i.e. data1 and data2), the Feature Selection stage deletes all unnecessary columns and produces the following tables:

	MovieID	Title	Genres
0	1	Toy Story (1995)	Animation Children's Comedy
1	2	Jumanji (1995)	Adventure Children's Fantasy
2	3	Grumpier Old Men (1995)	Comedy Romance
3	4	Waiting to Exhale (1995)	Comedy Drama
4	5	Father of the Bride Part II (1995)	Comedy

Figure 6.1 data1 dataset after Feature Selection Stage

	UserID	MovieID	Rating
0	1	1193	5
1	1	661	3
2	1	914	3
3	1	3408	4
4	1	2355	5

Figure 6.2 data2 dataset after Feature Selection Stage

6.1.2. Data Integration Stage Result

After all the unnecessary columns are removed, now the Data Integration stage starts to merge the two tables. The following figure shows the result of Data Integration stage:

	UserID	MovieID	Rating	Title	Genres
0	1.0	1193	5.0	One Flew Over the Cuckoo's Nest (1975)	Drama
1	2.0	1193	5.0	One Flew Over the Cuckoo's Nest (1975)	Drama
2	12.0	1193	4.0	One Flew Over the Cuckoo's Nest (1975)	Drama
3	15.0	1193	4.0	One Flew Over the Cuckoo's Nest (1975)	Drama
4	17.0	1193	5.0	One Flew Over the Cuckoo's Nest (1975)	Drama

Figure 6.3 data dataset after Data Integration Stage

6.1.3. Data Cleaning Stage Result

After the two independent tables are merged, the Data Cleaning stage removes all missing value rows that reduces the quality of the dataset. Before the Data Cleaning stage, the number of tuples is 1,000,386 and after the Data Cleaning stage i.e. after deleting all rows with missing values the number of tuples is 1,000,209.

6.1.4. Data Conversion Stage Result

Finally, before the k-means clustering algorithm starts, the Genre column should have to be converted to numerical, because the k-means algorithm only uses numerical data to calculate distance between items or movies. To change the Genre column to numerical one way is to convert all Genre values as columns and all the movies that have that Genre type is valued as 1, the rest valued as 0. The following figure shows the result of Data Conversion stage i.e. conversion of Genre column values to numerical value columns, initially the values of converted Genres is initialized as 0:

	UserID	MovieID	Rating	Title	Genres	Drama	Animation	Musical	Action	Comedy	...	Thriller	Crime	Western	Documentary	Mystery	Horror	Sci-Fi	Film
0	1.0	1193	5.0	One Flew Over the Cuckoo's Nest (1975)	Drama	0	0	0	0	0	...	0	0	0	0	0	0	0	0
1	2.0	1193	5.0	One Flew Over the Cuckoo's Nest (1975)	Drama	0	0	0	0	0	...	0	0	0	0	0	0	0	0

Figure 6.4 data dataset after Data Conversion Stage

After the Genres are created as columns, each column is initialized as 1 which is present in the Genres. The following picture shows the result of the dataset after the initialization of 1 for Genres:

	UserID	MovieID	Rating	Title	Genres	Drama	Animation	Musical	Action	Comedy	...	Thriller	Crime	Western	Documentary	Mystery	Horror	Sci-Fi	Film
0	1.0	1193	5.0	One Flew Over the Cuckoo's Nest (1975)	Drama	1	0	0	0	0	...	0	0	0	0	0	0	0	0
1	2.0	1193	5.0	One Flew Over the Cuckoo's Nest (1975)	Drama	1	0	0	0	0	...	0	0	0	0	0	0	0	0

Figure 6.5 data dataset after Data Conversion Stage

The Data Conversion Stage produces 23 columns and all the columns are not numerical which makes it difficult for K-means Clustering algorithm to perform distance calculation since K-means Clustering algorithm only works only on numerical dataset. Because of this, both Title and Genres columns are needed for distance calculation, then both these columns are deleted. The following figure shows the final result of Data Pre-processing stage:

	UserID	MovieID	Rating	Drama	Animation	Musical	Action	Comedy	Adventure	Romance	...	Thriller	Crime	Western	Documentary	Mystery	Horror	Sci-Fi
0	1.0	1193	5.0	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0
1	2.0	1193	5.0	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0
2	12.0	1193	4.0	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0
3	15.0	1193	4.0	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0
4	17.0	1193	5.0	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0

Figure 6.6 data dataset after Data Pre-processing Stage

6.2. Model (K-means and Hybrid PSO-KM) Evaluation Result

In this part the result of both K-means and Hybrid PSO-KM cluster results are compared and analyzed. First, the result of the K-means algorithm will be discussed, second, the result of the Hybrid PSO-KM algorithm will also be discussed. Finally, the result of Popularity-based Recommendation algorithm will be discussed. To evaluate the quality of cluster produced by the two algorithms, the following methods will be used: intra-cluster distance (Sum-of Squares-Error (SSE)) to determine the distance of items in cluster from the centroid, inter-cluster distance (Silhouette score) to determine the distance of neighboring clusters from each other and fitness of algorithm (Quantization Error) to determine which algorithm minimizes the average (squared) quantization error.

6.2.1. K-means Clustering Algorithm Result

After all the data pre-processing is completed and after the number of Cluster K is determined, the K-mean algorithm will be ready to create clusters. To create clusters that contain homogenous items this algorithm uses Euclidean distance calculation to find the correct centroids and reassign items in their closest clusters. The Euclidean distance calculation for finding a new centroid point and the reassignment of items to the closest cluster is performed iteratively until all elements are in their closest cluster.

Determining K (Number of Clusters)

K-Means is relatively an efficient method, but the final results of clusters are sensitive to initialization or number of clusters selected (K) and often terminates at a local optimum. To produce a quality cluster either the user iteratively selects a random number and determines which number helps K-means clustering to produce quality clusters or the user must use well known methods which determine the possible number that help K-means Clustering algorithm to produce quality clusters. The latter idea is better regarding time and efficiency. In this paper, the Elbow Method is used to determine the number of clusters (K) by comparing results across different values of K is the mean distance between data points and their cluster centroid. Since increasing the number of clusters will always reduce the distance to data points, increasing K will *always* decrease this metric, to the extreme of reaching zero when K is the same as the number of data points. Thus, this metric cannot be used as the sole target. Instead, mean distance to the centroid

as a function of K is plotted and the "elbow point," where the rate of decrease sharply shifts, can be used to roughly determine K . the following figure shows the result of Elbow Method:

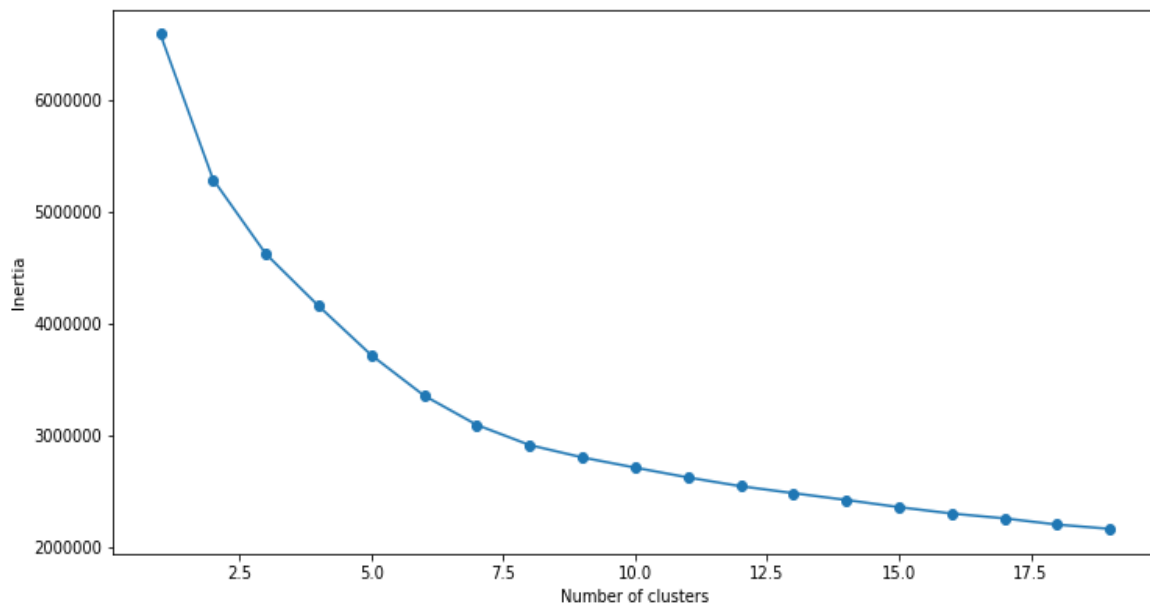


Figure 6.7 The Result of Elbow Method

By carefully studying the graph, the point or the elbow point where the rate decrease of mean distance to the centroid as a function of K sharply shifts is used to roughly determine K . The following figure shows the possible elbow point that can be used as K (number of cluster) i.e. 7 or 8 roughly.

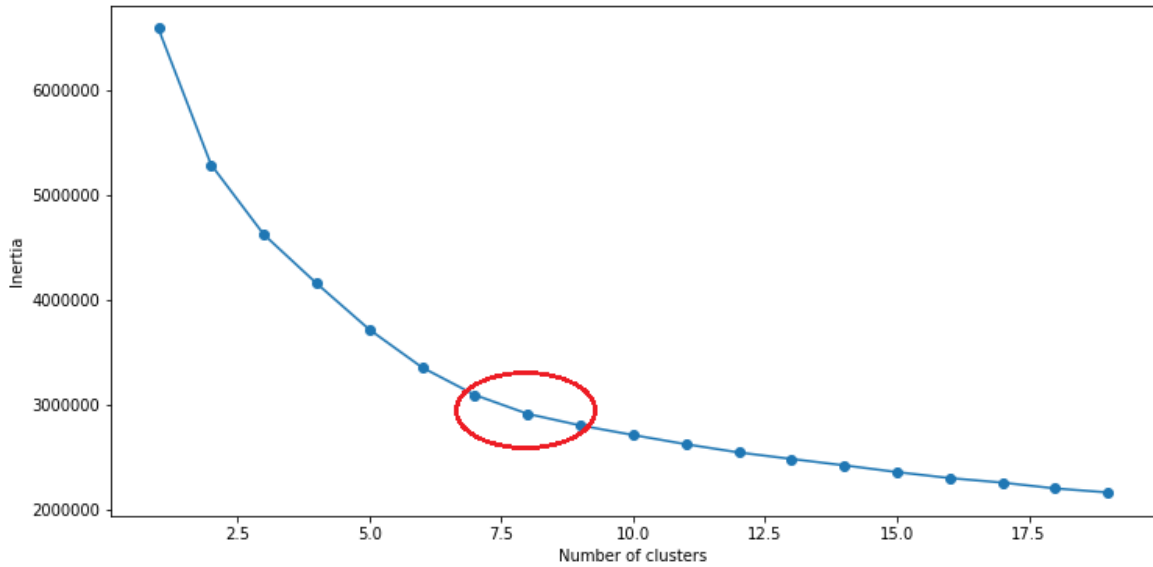


Figure 6.8 The Result of Elbow Method

In this paper, it is possible to use 7 or 8 as the number of clusters (K) for the K-means Clustering algorithm to create clusters. Since, the rate decrease of mean distance to the centroid as a function of K sharply shifts starts roughly around 8, 8 is used as a final value of k (number of clusters).

K-means Clustering Algorithm Results

After all the data pre-processing is completed and the number of Cluster (K) is determined, K-mean algorithm is now ready to find the correct centroids and reassign items in their closest clusters and also, to create quality clusters or homogenous clusters by iteratively calculating Euclidean distance calculation. The following figure shows the result of K-means Clustering after producing 8 homogenous clusters:

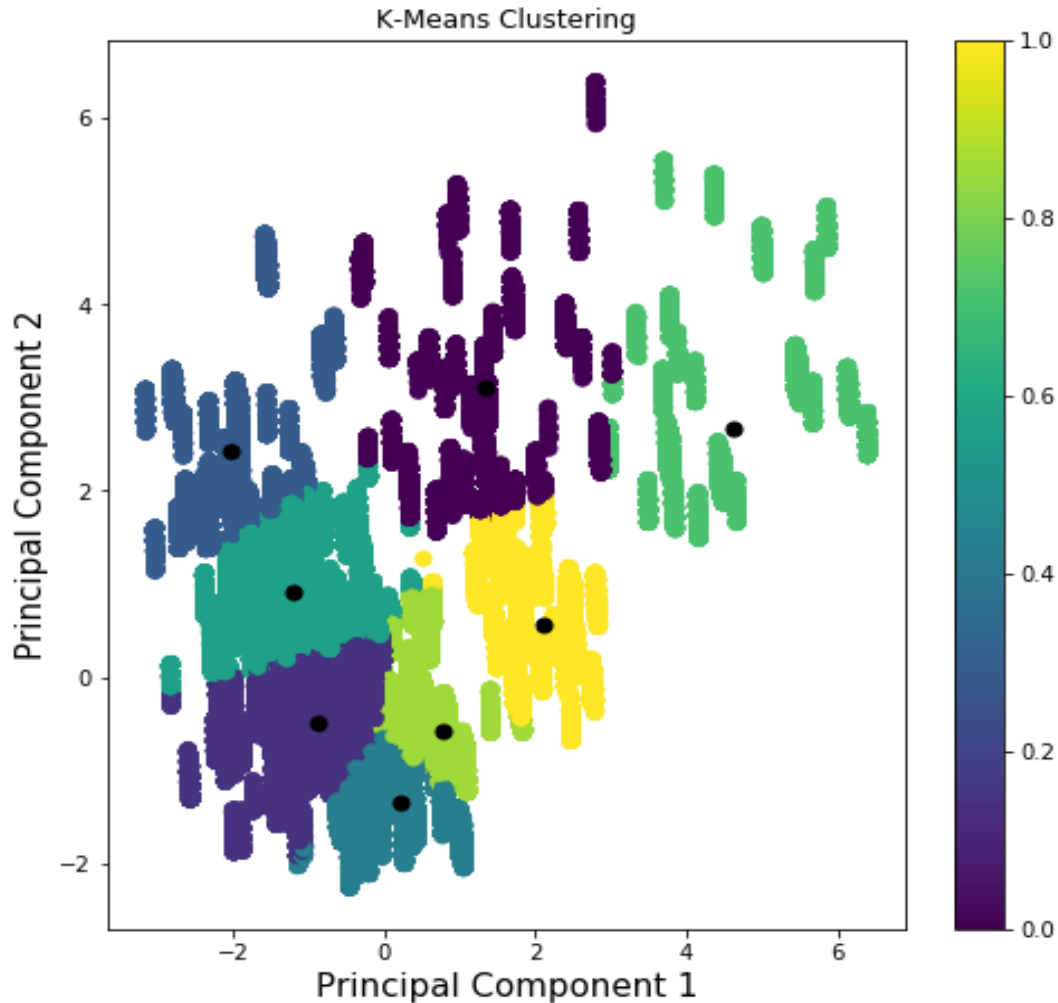


Figure 6.9 The Result of K-means Clustering

After the K-means Clustering algorithm finishes creating clusters, now the quality of clusters created is evaluated by using intra-cluster distance (Sum-of Squares-Error (SSE)) to determine the distance of items in cluster from the centroid, inter-cluster distance (Silhouette score) to determine the distance of neighboring clusters from each other and fitness of algorithm (Quantization Error) to determine which algorithm minimizes the average (squared) quantization error. The range of silhouette score (inter-cluster distance) is from -1 to 1. The better result of silhouette score is positive or closer to 1, which depicts the neighboring clusters are farthest to each other. Sum-of Squares-Error (SSE) determines from two algorithms which algorithm produces compact clusters. The smallest the distance the better the cluster quality is. The fitness of algorithm (Quantization Error) determines from two algorithms which algorithm has good fitness to produce quality clusters. The smaller the result the better fitness the algorithm has.

The following figure shows the Evaluation results of K-means Clustering Algorithm:

```
Silhouette: 0.20670296703922036  
SSE: 1117770.360384986  
Quantization: 343.5844860895621
```

Figure 6.10 The Result of K-means Clustering Algorithm Evaluation

The Evaluation results of K-means Clustering Algorithm will be used to determine and compare the result of cluster quality with the Evaluation results of Hybrid PSO-KM Algorithm.

6.2.2. Hybrid PSO-KM (Particle Swarm Optimization and K-means Clustering) Algorithm Result

By combining PSO and K-means (PSO-KM), first by using the Elbow Method the number of cluster K is determined, then the PSO algorithm searches all space for a global solution. When PSO algorithm finishes by finding the region of global optimum solution iteratively, then, k-means algorithms start clustering to refine clusters to finalize the clustering task by finding local optimum solution iteratively as indicated in section 5.6. This hybrid algorithm accelerates the convergence speed as well as accuracy by using the PSO algorithm to discover the vicinity of the optimal solution by a global search and the result of PSO is used as the initial seed of the K-means algorithm. Finally, K-means algorithm refine clusters and generate the final result.

For implementing Hybrid PSO-KM algorithm the following parameters are used: for the better result PSO-KM algorithm by repetitively testing the number of particles used to search the problem space by varying different numbers, the number of particles recommended is 10. The maximum iteration allowed for this algorithm is 2,000 times and the number of clusters used is 8, which is determined by the Elbow method as indicated in section 6.2.1.

The following figure shows the result of Hybrid PSO-KM algorithm after producing 8 homogenous clusters:

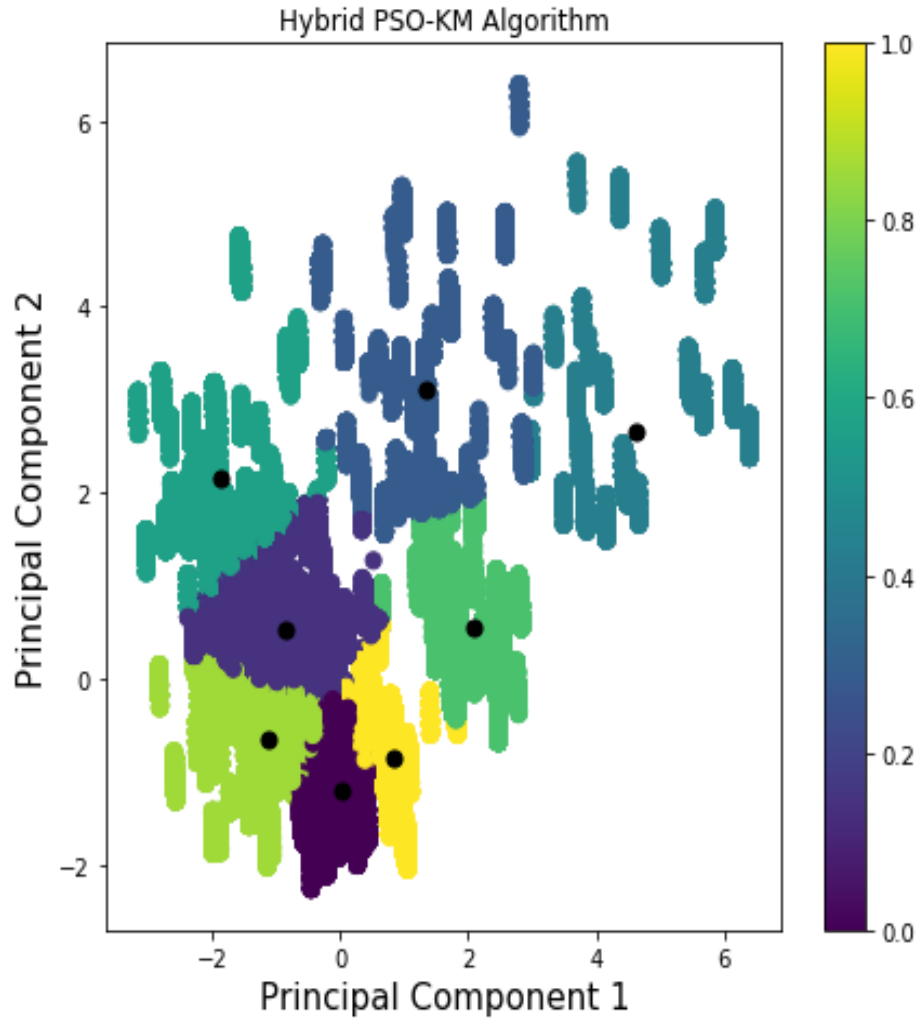


Figure 6.11 The Result of Hybrid PSO-KM Algorithm Evaluation

After the Hybrid PSO-KM algorithm finishes creating clusters, now the quality of clusters created is evaluated by using intra-cluster distance (Sum-of Squares-Error (SSE)) to determine the distance of items in cluster from the centroid, inter-cluster distance (Silhouette score) to determine the distance of neighboring clusters from each other and fitness of algorithm (Quantization Error) to determine which algorithm minimizes the average (squared) quantization error. The range of silhouette score (inter-cluster distance) is from -1 to 1. The better result of silhouette score is positive or closer to 1, which depicts the neighboring clusters are farthest to each other. Sum-of Squares-Error (SSE) determines from two algorithms which algorithm produces compact clusters. The smallest the distance the better the cluster quality is. The fitness of algorithm (Quantization

Error) determines from two algorithms which algorithm has good fitness to produce quality clusters. The smaller the result the better fitness the algorithm has.

The following figure shows the Evaluation results of Hybrid PSO-KM Algorithm:

```
Silhouette: 0.20472528090633088
SSE: 1107488.668556686
Quantization: 352.1225031538951
```

Figure 6.12 The Result Hybrid PSO-KM Algorithm Evaluation

The Evaluation results of Hybrid PSO-KM Algorithm will be used to determine and compare the result of cluster quality with the Evaluation results of K-means Clustering Algorithm.

6.2.3. Popularity-based Recommendation Algorithm Result

Recommendation systems are one of helper systems that search through all possible options and create a personalized list of items that are interesting and relevant to an individual.

In popularity-based Recommendation Algorithm, first, by accepting the user_ID and cluster number the given user belongs, this algorithm computes the following: First, for all movies in the dataset average ratings of each movie is calculated and this is used in selecting popular movies in the given cluster. The following figure shows the dataset with average movie ratings for each move and with Title merged with data dataset:

Animation	Musical	Action	Comedy	Adventure	Romance	...	Documentary	Mystery	Horror	Sci-Fi	Horror	War	Fantasy	Cluster	Title	AvgMovieRating
0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	One Flew Over the Cuckoo's Nest (1975)	4.390725
0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	One Flew Over the Cuckoo's Nest (1975)	4.390725
0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	One Flew Over the Cuckoo's Nest (1975)	4.390725
0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	One Flew Over the Cuckoo's Nest (1975)	4.390725

Figure 6.13 The Result data dataset after average movie rating and Title are included

Second, all the movies the given user didn't see or rate before in the given cluster is collected. From this collected list the popular movies in the given cluster is sorted in descending order by

using average movie rating calculated earlier. The following figure shows the dataset after all the unique movies from given cluster the given user didn't rate are selected:

Animation	Musical	Action	Comedy	Adventure	Romance	...	Documentary	Mystery	Horror	Sci-Fi	Crime-Noir	War	Fantasy	Cluster	Title	AvgMovieRating
0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	Erin Brockovich (2000)	3.863878
1.0	0.0	0.0	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	Bug's Life, A (1998)	3.854375
0.0	0.0	0.0	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	Christmas Story, A (1983)	4.238905
0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	Miracle on 34th Street (1947)	3.960526
0.0	0.0	0.0	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	Ferris Bueller's Day Off (1986)	4.117447

Figure 6.14 The Result data dataset after all the movies the user didn't rate are selected

Finally, the algorithm recommends to the given user the top 20 or 30 movies in the top of sorted popular movies. For demonstration purposes the following parameters are used: UserID=10, cluster=2. The following figure shows the result of popularity-based recommendation algorithm (top 20 movies in that cluster based on average movie rating) by sorting movies based on average movie rating and selecting the top popular 20 movies for given user:

Title	
AvgMovieRating	
5.000000	Baby, The (1973)
4.800000	I Am Cuba (Soy Cuba/Ya Kuba) (1964)
4.666667	Apple, The (Sib) (1998)
4.608696	Sanjuro (1962)
4.560510	Seven Samurai (The Magnificent Seven) (Shichin...
4.500000	Skipped Parts (2000)
4.500000	Bells, The (1926)
4.444444	For All Mankind (1989)
4.415608	Double Indemnity (1944)
4.406263	Sixth Sense, The (1999)
4.404651	Yojimbo (1961)
4.400000	Return with Honor (1998)
4.387454	City Lights (1931)
4.368932	General, The (1927)
4.357143	Bicycle Thief, The (Ladri di biciclette) (1948)
4.339394	Grand Illusion (Grande illusion, La) (1937)
4.337354	Saving Private Ryan (1998)
4.335766	Creature Comforts (1990)
4.333333	Soft Fruit (1999)
4.329861	Life Is Beautiful (La Vita è bella) (1997)

Figure 6.15 The Top 20 Movies Recommended by Popularity-based Recommendation Algorithm

6.3. Discussion

This section compares the results of the K-means and Hybrid clustering. The main purpose is to compare the quality of the respective clustering, where quality is measured according to the following three criteria:

- the Quantization Error, i.e. indicates the value of the average distance between items with cluster centroid to which they belong. The smaller the Quantization Error value, the more compact the clustering solution is and the better fitness the algorithm has.
- the Intra-cluster Distances (Sum-of Squares-Error (SSE)), i.e. the distance between items within a cluster, where the objective is to minimize the SSE, the smaller the SSE the better the quality of the clusters the algorithm produces;
- the Inter-cluster Distances (Silhouette Score), i.e. the distance between the centroids of the clusters, where the objective is to maximize the distance between clusters. the smaller the Silhouette Score the better the quality of the clusters the algorithm produces.

In this paper, to implement and compare both K-means Clustering and Hybrid PSO-KM algorithms the following parameters are used: the number of particles used is 10. The maximum iteration allowed for this algorithm is 2,000 times and the number of clusters used is 8. And the dataset used for training both algorithms is MovieLens (ml-latest-small) dataset. And also, for comparison purposes both 7 and 8 clusters are used to compare and contrast the result of the two algorithms.

The following tables below show the performances of both K-means Clustering and Hybrid PSO-KM algorithms regarding which algorithm produces better quality clusters.

Table 6.1 Comparisons of K-means Algorithm and Hybrid PSO-KM Algorithm using 7 clusters

No	Algorithms	SSE	Silhouette Score	Quantization Error
1	K-means Algorithm	1139116.4668380944	0.19228897420414753	378.24970250671424

2	Hybrid PSO-KM Algorithm	1103346.221883 0157	0.18500490628647 148	392.9875600678043
---	-------------------------	------------------------	-------------------------	-------------------

As shown in above Table 6.1. the Inter-cluster Distances (Silhouette Score) which is the distance between the centroids of the clusters or the distance between the neighboring clusters, Hybrid PSO-KM Algorithm achieves a better separation between the neighboring clusters than K-means Algorithm. The Hybrid PSO-KM Algorithm produces 3.94% better separation between the neighboring clusters than the K-means Algorithm. Regarding Sum-of Squares-Error (SSE), which is the distance between items within a cluster and the smaller the SSE the compact the items in a given cluster. Based on the above table, the Hybrid PSO-KM Algorithm produces more compact results than K-means Algorithm. The Hybrid PSO-KM Algorithm produces 3.24% better or compact clusters than the K-means algorithm. And finally, regarding Quantization Error, which is the average distance between items with cluster centroid to which they belong, the K-means algorithm has better results than Hybrid PSO-KM Algorithm. K-means algorithm produces 3.9% better results that reduce the average distance between items with cluster centroid in the same cluster than Hybrid PSO-KM Algorithm

Table 6.2 Comparisons of K-means Algorithm and Hybrid PSO-KM Algorithm using 8 clusters

No.	Algorithms	SSE	Silhouette Score	Quantization Error
1	K-means Algorithm	1117770.3603 84986	0.206702967039 22036	343.5844860895621
2	Hybrid PSO-KM Algorithm	1107488.6685 56686	0.204725280906 33088	352.1225031538951

As shown in above Table 6.2. the Inter-cluster Distances (Silhouette Score) which is the distance between the centroids of the clusters or the distance between the neighboring clusters, Hybrid PSO-KM Algorithm achieves a better separation between the neighboring clusters than K-means

Algorithm. The Hybrid PSO-KM Algorithm produces 0.97% better separation between the neighboring clusters than the K-means Algorithm. Regarding Quantization Error, which is the average distance between items with cluster centroid to which they belong, the K-means algorithm has better results than Hybrid PSO-KM Algorithm. K-means algorithm produces 2.48% better results that reduce the average distance between items with cluster centroid in the same cluster than Hybrid PSO-KM Algorithm. And finally, regarding Sum-of Squares-Error (SSE), which is the distance between items within a cluster and the smaller the SSE the compact the items in a given cluster. Based on the above table, the Hybrid PSO-KM Algorithm produces more compact results than K-means Algorithm. The Hybrid PSO-KM Algorithm produces 0.93% better or compact clusters than the K-means algorithm.

Table 6.3 The comparisons of results of the two algorithms using 7 and 8 clusters

No.	Algorithms	7 Clusters			8 Clusters		
		SSE	Silhouette Score	Quantization Error	SSE	Silhouette Score	Quantization Error
1	K-means Algorithm	11391 16.466 83809 44	0.18500490 628647148	392.987560067 8043	111777 0.3603 84986	0.2067029 67039220 36	343.5844860 895621
2	Hybrid PSO-KM Algorithm	11033 46.221 88301 57	0.19228897 420414753	378.249702506 71424	110748 8.6685 56686	0.2047252 80906330 88	352.1225031 538951

Finally, by seeing the above Table 6.3. and by considering the above parameters used, it is possible to conclude that, in both 7 and 8 clusters Hybrid PSO-KM Algorithm achieves by producing compact clusters than K-means algorithm and also produced better separation between the neighboring clusters than K-means Algorithm. When clusters are 7 and 8, the K-means Algorithm produced result that reduce the average distance between items with cluster centroid in the same cluster than Hybrid PSO-KM Algorithm. From this it is possible to conclude that, the Hybrid PSO-

KM Algorithm achieves both in intra-cluster distance and inter-cluster distance than the K-means algorithm by producing compact and better separation between neighboring clusters than the K-means Algorithm. The K-means algorithm achieves the fitness function better than Hybrid PSO-KM Algorithm. Finally, by seeing all the above results, it is also possible to say that Hybrid PSO-KM Algorithm produced better compact and quality clusters than K-means Clustering Algorithm.

Chapter Seven

7. Conclusion, Recommendation and Future Work

7.1. Conclusion

In this paper, the application of both K-means and PSO algorithms used to compare the result of K-means with Hybrid PSO-KM Algorithm by creating Hybrid algorithm by using both K-means and PSO algorithms. For experimenting these algorithms MovieLens (ml-latest-small) dataset is used. First, before starting the model the above dataset passed through successive stages of Data Pre-processing to clean unnecessary columns and remove missing value rows and also, to prepare quality numerical data that is ready to train the model. The k-means clustering algorithm is trained on quality data and produces homogenous clusters and the results of the algorithm is evaluated by using SSE, Silhouette Score and Quantization Score methods. The Hybrid PSO-KM Algorithm is trained with quality data and produced homogenous clusters. First, the PSO algorithm produced a global optimal solution that was used by the K-means algorithm to find a local optimal solution. Finally, the result of Hybrid PSO-KM Algorithm is evaluated by using SSE, Silhouette Score and Quantization Score methods. After completing creating quality and homogenous clusters that ensures that most users who have similar movie taste are clustered in the same clusters. The popularity-based recommendation algorithm recommended the top 20 movies which the user never rated before based on their average movie ratings.

By using the above-mentioned cluster evaluation methods i.e. SSE, Silhouette Score and Quantization Score methods both algorithms are evaluated. The aim of proposing to use Hybrid PSO-KM Algorithm is to improve the drawback of both algorithms and to produce more compact and quality clusters. The evaluation results confirm that in both 7 and 8 clusters Hybrid PSO-KM Algorithm achieves by producing compact clusters than K-means algorithm and also produced better separation between the neighboring clusters than K-means Algorithm. But when clusters are 7 and 8, the K-means Algorithm produced result that reduce the average distance between items with cluster centroid in the same cluster than Hybrid PSO-KM Algorithm. From this it is possible to conclude that, the Hybrid PSO-KM Algorithm achieves both in intra-cluster distance and inter-cluster distance than the K-means algorithm by producing compact and better separation between neighboring clusters than the K-means Algorithm.

7.2. Recommendation

Since Recommendation Algorithms are very important in customer-centric companies and systems in helping users to search and filter the goods or items they are looking for quickly. It also helps companies in minimizing find-sale users' transactions which helps companies to serve more users and to have satisfied customers who find what they are searching for. [In Ethiopia, Recommender systems can be applied in businesses like Hotels and Restaurants Recommendation, Books Recommendation, Tourism places Recommendation and also Jobs Recommendation.](#) Because of vast benefits of Recommender systems, our country customer-centric companies must start to build and integrate Recommender systems to their services. Both the companies and users will be benefited by it greatly.

7.3. Future Work

In this paper, by using MovieLens (ml-latest-small) dataset the application of both K-means and PSO algorithms used to compare the result of K-means with Hybrid PSO-KM Algorithm. The number of features used from MovieLens (ml-latest-small) dataset is four i.e. user_id, movie_id, genres and user rated movies or rating features. The Elbow Method is used to determine the number of clusters K which promises to produce optimal cluster results.

Next, in recommendation systems to improve the recommendation systems' ability to recommend user-related goods or items, training the algorithm with more users personal and social data improves the efficiency of recommendation systems. And also, training the machine learning algorithm with more data can also improve the performance of recommendation algorithms.

References

- [1] J. A. Konstan and J. Riedl, "Recommender systems: From algorithms to user experience," *User Model. User-adapt. Interact.*, vol. 22, no. 1–2, pp. 101–123, 2012.
- [2] R. Burke, "Hybrid Recommender Systems: Survey and," *User Model. User-Adapted Interact. J. Pers. Res.*, vol. 12, no. C, pp. 2005–2008, 2008.
- [3] R. Hu and P. Pu, "Acceptance issues of personality-based recommender systems," *ACM New York, USA* p. 221, 2009.
- [4] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," *Microsoft Research, USA*, pp. 43–52, 2013.
- [5] F. Ricci, L. Rokach, and B. Shapira, "Recommender Systems Handbook" *Springer Science, Business Media*, 2015.
- [6] F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh, "Recommendation systems: Principles, methods and evaluation," *Egypt. Informatics J.*, vol. 16, no. 3, pp. 261–273, 2015.
- [7] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, "Recommender system application developments: A survey," *Decis. Support Syst.*, vol. 74, pp. 12–32, 2015.
- [8] R. Burke, "Hybrid Web Recommender Systems," pp. 377–408, 2007.
- [9] D. Marutho, S. Hendra Handaka, E. Wijaya, and Muljono, "The Determination of Cluster Number at k-Mean Using Elbow Method and Purity Evaluation on Headline News," *Proc. - 2018 Int. Semin. Appl. Technol. Inf. Commun. Creat. Technol. Hum. Life, iSemantic 2018*, pp. 533–538, 2018.
- [10] Trupti M. Kodinariya, Dr. Prashant R. Makwana, "Review on Determining of Cluster in K-means Clustering Review on determining number of Cluster in K-Means Clustering" *International Journal of Advance Research in Computer Science and Management Studies* 2013 vol. 1, no. July, pp. 90–95, 2016.
- [11] A. E. DW van der Merwe, "Data Clustering using Particle Swarm Optimization," *IEEE*, pp. 15–220, 2003.

- [12] X. Cui, T. E. Potok, and P. Palathingal, "Document Clustering using Particle Swarm Optimization," *IEEE*, pp. 1–7, 2005.
- [13] GroupLens, "MovieLens Dataset," 2020. [Online]. Available: <https://grouplens.org/datasets/movielens/>. [Accessed: 06-Nov-2020].
- [14] M. NK, "K-Means Clustering in Python," 2020. [Online]. *mubaris.com*. [Accessed: 10-Nov-2020].
- [15] G. Saini and H. Kaur, "K-Mean Clustering and PSO: A Review," *Blue Eyes Intelligence Engineering & Sciences Publication Pvt. Ltd* no. 5, pp. 112–114, 2014.
- [16] O. Ismael, "A new hybrid approach for document clustering," *ICENCO 2017 - 13th Int. Comput. Eng. Conf. Boundless Smart Soc.*, vol. 2018-Janua, pp. 291–295, 2018.
- [17] D.-A. Manolescu, M. Voelter, and J. Noble, "Data conversion." Addison-Wesley, p. 596, 2006.
- [18] Techopedia, "What is Data Validation? - Definition from Techopedia," *Techopedia*, 2019.
- [19] S. S. S. Binitha S, "A Survey of Bio inspired Optimization Algorithms," *Int. J. Soft Comput. Eng.*, vol. 2, no. 2, pp. 2323–2339, 2012.
- [20] B. Pathak, R. Garfinkel, R. D. Gopal, R. Venkatesan, and F. Yin, "Empirical Analysis of the Impact of Recommender Systems on Sales," *J. Manag. Inf. Syst.*, vol. 27, no. 2, pp. 159–188, 2010.
- [21] S. Bhat, D. Dharrao, V. Pingale, S. Agarwal, and S. Bhadauriya, "JASC : Journal of Applied Science and Computations ISSN NO : 1076-5131 Automated Online Parking Systems Page No : 1479," vol. V, no. 1478, pp. 1478–1483, 2018.
- [22] I. No, "Available Online at www.ijarcs.info International Journal of Advanced Research in Computer Science A Review on MANET and DTN for Hybrid Network Architecture," vol. 5, no. 7, pp. 72–75, 2014.
- [23] G. Jawaheer, M. Szomszor, and P. Kostkova, "Comparison of implicit and explicit feedback from an online music recommendation service," *ACM*, pp. 47–51, 2010.
- [24] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative Filtering for Implicit Feedback Datasets

- Yifan,” *Gastroenterology*, vol. 1), p. S415, 2015.
- [25] G. A. Tsihrintzis and D. N. Sotiropoulos, *Intelligent Systems Reference Library 149 Machine Learning Paradigms*. 2018.
 - [26] C. Troussas, A. Krouska, and M. Virvou, “Multi-algorithmic techniques and a hybrid model for increasing the efficiency of recommender systems,” *Proc. - Int. Conf. Tools with Artif. Intell. ICTAI*, vol. 2018-Novem, pp. 184–188, 2018.
 - [27] M. S. B. Praba, “Educational Website Based on Dynamic Recommendation and Filtering Techniques,” vol. 6, no. 10, pp. 84–87, 2018.
 - [28] A. Davoudi and M. Chatterjee, “Social trust model for rating prediction in recommender systems: Effects of similarity, centrality, and social ties,” *Online Soc. Networks Media*, vol. 7, pp. 1–11, 2018.
 - [29] X. Huang, X. Yang, J. Zhao, L. Xiong, and Y. Ye, “A new weighting k-means type clustering framework with an l2-norm regularization,” *Knowledge-Based Syst.*, vol. 151, pp. 165–179, 2018.
 - [30] B. D. Ripley and M. Thompson, “Regression techniques for the detection of analytical bias,” *Analyst*, vol. 112, no. 4, pp. 377–383, 1987.
 - [31] M. Jahrer, A. Töschler, and R. Legenstein, “Combining Predictions for an accurate Recommender System.”
 - [32] I. A. Basheer and M. Hajmeer, “Artificial neural networks : fundamentals , computing , design , and application,” *J. Microbiol. Methods*, vol. 43, pp. 3–31, 2000.
 - [33] A. Benbassat and M. Sipper, “Evolving artificial neural networks with FINCH,” *IEEE*, vol. 87, no. 9, p. 1719, 2013.
 - [34] K. Yu, A. Schwaighofer, V. Tresp, X. Xu, and H. Kriegel, “Probabilistic Memory-Based Collaborative Filtering,” *IEEE*, vol. 16, no. 1, pp. 56–69, 2004.
 - [35] D. M. Pennock, E. Horvitz, S. Lawrence, and C. L. Giles, “Collaborative filtering by personality diagnosis: A hybrid memory-and model-based approach,” *Proc. 16th Conf. Uncertain. Artif. Intell.*, vol. 64, no. 10, pp. 473–480, 2000.

- [36] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," *Proc. 10th ...*, vol. 1, pp. 285–295, 2001.
- [37] R. Burke, "Knowledge-based recommender systems," pp. 1–23, 1999.
- [38] from Techopedia.com, "Data Preprocessing," *from Techopedia*. [Online]. Available: <https://www.techopedia.com/definition/14650/data-preprocessing>. [Accessed: 18-Jul-2019].
- [39] the free encyclopedia From Wikipedia, "Data pre-processing," *the free encyclopedia*. [Online]. Available: https://en.wikipedia.org/wiki/Data_pre-processing. [Accessed: 18-Jul-2019].
- [40] M. Rajaratne, "Data Pre-Processing Techniques You Should Know," *from towardsdatascience.com*. [Online]. Available: <https://towardsdatascience.com/data-pre-processing-techniques-you-should-know-8954662716d6>. [Accessed: 18-Jul-2019].
- [41] T. Wangchamhan, S. Chiewchanwattana, and K. Sunat, "Efficient algorithms based on the k-means and Chaotic League Championship Algorithm for numeric, categorical, and mixed-type data clustering," *Expert Syst. Appl.*, vol. 90, pp. 146–167, 2017.
- [42] N. Singh, "Overview of Encoding Methodologies." [Online]. Available: <https://www.datacamp.com/community/tutorials/encoding-methodologies>. [Accessed: 18-Jul-2019].
- [43] A. Ahmad and S. Hashmi, "K-Harmonic means type clustering algorithm for mixed datasets," *Appl. Soft Comput. J.*, vol. 48, pp. 39–49, 2016.
- [44] J. Ji, T. Bai, C. Zhou, C. Ma, and Z. Wang, "Neurocomputing An improved k-prototypes clustering algorithm for mixed numeric and categorical data," *Neurocomputing*, vol. 120, pp. 590–596, 2013.
- [45] T. R. L. Dos Santos and L. E. Zárate, "Categorical data clustering: What similarity measure to recommend?" *Expert Syst. Appl.*, vol. 42, no. 3, pp. 1247–1260, 2015.
- [46] B. Ben Ali, "K-Means clustering based on Gower Similarity Coefficient: A comparative study," *IEEE*, 2013.

- [47] A. Ahmad and L. Dey, “A k-mean clustering algorithm for mixed numeric and categorical data,” *Data Knowl. Eng.*, vol. 63, no. 2, pp. 503–527, 2007.
- [48] S. Shalev-Shwartz and S. C. N.-Q... S. 2014 Ben-David, *Understanding machine learning: from theory to algorithms*. 2014.
- [49] H. D. III, “A course in Machine Learning” *Hal Daume III*, August 2012.
- [50] MathWorks, “Introducing Machine Learning What is Machine”, *MathWorks*, January 2016.
- [51] M. Learning *et al.*, “Machine learning” *M.L.* 2015.
- [52] O. Chapelle, B. Schölkopf, and A. Zien, “Semi-supervised learning,” 2006.
- [53] S. Ben Salem, S. Naouali, and Z. Chtourou, “A fast and effective partitional clustering algorithm for large categorical datasets using a k-means based approach,” *Comput. Electr. Eng.*, vol. 68, no. May, pp. 463–483, 2018.
- [54] T. Zeugmann, H. University, and J. Sapparo, “PAC Identification,” *Encycl. Mach. Learn.*, vol. 1, p. 74, 2010.
- [55] Z. Huang and M. K. Ng, “A Note on K-modes Clustering,” *J. Classif.*, vol. 20, no. 2, pp. 257–261, 2003.
- [56] A. Bandit, “Encyclopedia of Machine Learning and Data Mining,” *Encycl. Mach. Learn. Data Min.*, 2015.
- [57] X. S. Yang, *Nature-Inspired Optimization Algorithms*. 2014.
- [58] S. A. Uymaz, G. Tezel, and E. Yel, “Artificial algae algorithm (AAA) for nonlinear global optimization,” *Appl. Soft Comput. J.*, vol. 31, pp. 153–171, 2015.
- [59] A. Ahmadyfard and H. Modares, “Combining PSO and k-means to Enhance Data Clustering,” pp. 688–691, 2008.
- [60] T. Cura, “Expert Systems with Applications A particle swarm optimization approach to clustering,” *Expert Syst. Appl.*, vol. 39, no. 1, pp. 1582–1588, 2012.
- [61] D.-A. Manolescu, M. Voelter, and J. Noble, “Data conversion.” Addison-Wesley, p. 596, 2006, Accessed: Apr. 22, 2019. [Online]. Available:

https://en.wikipedia.org/wiki/Data_conversion.

- [62] C. C. Hsu, H. C. Chen, K. K. Huang, and Y. M. Huang, “A personalized auxiliary material recommendation system based on learning style on Facebook applying an artificial bee colony algorithm,” *Comput. Math. with Appl.*, vol. 64, no. 5, pp. 1506–1513, 2012, doi: 10.1016/j.camwa.2012.03.098.
- [63] G. Sumathi, S. Sendhilkumar, and G. S. Mahalakshmi, “Hybrid recommendation system using particle swarm optimization and user access-based ranking,” *ACM Int. Conf. Proceeding Ser.*, vol. 25-26-August-2016, 2016, doi: 10.1145/2980258.2980405.
- [64] X. Wang, F. Luo, Y. Qian, and G. Ranzi, “A personalized electronic movie recommendation system based on support vector machine and improved particle swarm optimization,” *PLoS One*, vol. 11, no. 11, pp. 1–17, 2016, doi: 10.1371/journal.pone.0165868.
- [65] C. Christakou and A. Stafylopatis, “A hybrid movie recommender system based on neural networks,” vol. 16, no. 5, pp. 771–792, 2007.
- [66] J. Bennett and S. Lanning, “The Netflix Prize,” *Proc. KDD Cup Work.*, pp. 3–6, 2007.
- [67] Raheel Shaikh, “Feature Selection Techniques in Machine Learning with Python,” *Towards DataScience*. .
- [68] the free encyclopedia From Wikipedia, “Feature selection,” *From Wikipedia, the free encyclopedia*. .
- [69] M. Brun *et al.*, “Model-based evaluation of clustering validation measures,” *Pattern Recognit.*, vol. 40, no. 3, pp. 807–824, 2007.
- [70] Deepak jain, “Data Preprocessing in Data Mining,” *GeeksForGeeks*. [Online]. Available: <https://www.geeksforgeeks.org/data-preprocessing-in-data-mining/>.
- [71] S. Rana, S. Jasola, and R. Kumar, “A hybrid sequential approach for data clustering using K-Means and particle swarm optimization algorithm A hybrid sequential approach for data clustering using K-Means and particle swarm optimization algorithm,” no. May 2014, 2010.
- [72] datanovia, “Cluster Validation Essentials,” *datanovia*, 2018. [Online]. Available: <https://www.datanovia.com/en/lessons/determining-the-optimal-number-of-clusters-3->

must-know-methods/. [Accessed: 08-Mar-2019].

- [73] saedsayad, “K-Means Clustering,” *saedsayad.com*. [Online]. Available: https://www.saedsayad.com/clustering_kmeans.htm. [Accessed: 08-May-2019].
- [74] K. Dhanalakshmi, A. Anitha, G. Michael, and K. G. S. Venkatesan, “Recommendation System Based on Clustering and Collaborative Filtering,” pp. 2482–2488, 2015.
- [75] intranet.deib, “A Tutorial on Clustering Algorithms,” *Politecnico Milano 1863*. .
- [76] Z. A. Mur, “Numerical Example of k-means Clustering,” *revoledu.com*, 2017.

Appendixes

1. Appendix 1: Hybrid PSO-KM Algorithm and K-means Algorithm

```
#All the header files required for the Model
from pso import ParticleSwarmOptimizedClustering
from particle import quantization_error, calc_sse
from kmeans import KMeans
from sklearn.metrics import silhouette_score
from utils import normalize
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.sparse import csr_matrix
import helper
from factor_analyzer import FactorAnalyzer
from sklearn import metrics
import random
from utils import normalize
```

```
#All the header files required for the Model
from pso import ParticleSwarmOptimizedClustering
from particle import quantization_error, calc_sse
from kmeans import KMeans
from sklearn.metrics import silhouette_score
from utils import normalize
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.sparse import csr_matrix
import helper
from factor_analyzer import FactorAnalyzer
from sklearn import metrics
import random
from utils import normalize
```

```
#loading the datasets
data1 = pd.read_csv("ml-latest-small/ml small movies data/movies data.csv")
data2 = pd.read_csv("ml-latest-small/ml small movies data/ratings data.csv")
#Deleting unnecessary columns
data1 = data1.drop('Unnamed: 0',axis = 1)
data2 = data2.drop(['Unnamed: 0','Timestamp'],axis = 1)
#Merging or Integrating both the datasets
data = pd.merge(data2 , data1 , how='outer', on='MovieID')
#Merging or Integrating both the datasets
data = pd.merge(data2 , data1 , how='outer', on='MovieID')
#drop the null values coz the are not treatable
data.dropna(inplace= True )
```

```

# Converting Genres into different columns and initialize values as 0
x = data.Genres
a = list()
for i in x:
    abc = i
    a.append(abc.split('|'))
a = pd.DataFrame(a)
b = a[0].unique()
for i in b:
    data[i] = 0

# assign 1 to all the columns which are present in the Genres
for i in b:
    data.loc[data['Genres'].str.contains(i), i] = 1

# Deleting Genres and Title columns
data = data.drop(['Genres', 'Title'], axis = 1)
data.head()

data_norm = data
data_norm = data_norm.values
data_norm = normalize(data_norm)

# fitting multiple k-means algorithms and storing the values in an empty list
SSE = []
for k in range(1, 20):
    kmeans = KMeans(n_clusters = k)
    kmeans.fit(data)
    SSE.append(kmeans.inertia_)

# converting the results into a dataframe and plotting them
frame = pd.DataFrame({'Cluster': range(1, 20), 'SSE': SSE})
plt.figure(figsize=(12, 6))
plt.plot(frame['Cluster'], frame['SSE'], marker='o')
plt.xlabel('Number of clusters')
plt.ylabel('Inertia')

# k-means creating 8 homogenous clusters
kmeans = KMeans(n_clusters = 8)
#to create clusters
kmeans.fit(data_norm)
#to predict clusters
predicted_kmeans = kmeans.predict(data_norm)

User_label = kmeans.labels_
User_mean = kmeans.cluster_centers_

# Standardizing the features
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
y = StandardScaler().fit_transform(data_norm)
pca = PCA(n_components=2).fit(y)
principalComponents = pca.fit_transform(y)
principalDf = pd.DataFrame(data = principalComponents
                           , columns = ['principal component 1', 'principal component 2'])
principalDf.head()

```

```

# When dataframe is merged for a single movie multiple rows were created so a single movie is allotted
# assigning single cluster to a movie
e = []
def fi(group):
    a = pd.DataFrame(group)
    b = pd.DataFrame(a['Cluster'].value_counts())
    d = a.index
    c = [a['MovieID'][d[0]],int(b.idxmax())]
    e.append(c)

data.groupby("MovieID").apply(lambda x: fi(x))

#Hybrid PSO-KM implementation no particle=10
pso_rep = ParticleSwarmOptimizedClustering(
    n_cluster=8, n_particles=10, data=data_norm, hybrid=True,
    max_iter=2000, print_debug=2000)
pso_rep.run()
pso_kmeans = KMeans(n_cluster=8, init_pp=False)
pso_kmeans.centroid = pso_rep.gbest_centroids.copy()
predicted_pso_rep = pso_kmeans.predict(data_norm)

#Evaluating K-means Clustering
print('Silhouette:', silhouette_score(data_norm, predicted_kmeans))
print('SSE:', kmeans.SSE)
print('Quantization:', quantization_error(centroids=kmeans.centroid, data=data_norm, labels=predicted_kmeans))

def doKmeans(Hetrek_db, nclust=8):
    model = KMeans(nclust)
    model.fit(principalDf)
    clust_labels = model.predict(principalDf)
    cent = model.cluster_centers_
    return (clust_labels, cent)
clust_labels, cent = doKmeans(principalDf, 8)
kmeans = pd.DataFrame(clust_labels)

fig = plt.figure(figsize = (8,8))
ax = fig.add_subplot(1,1,1)
ax.set_xlabel('Principal Component 1', fontsize = 15)
ax.set_ylabel('Principal Component 2', fontsize = 15)
ax.set_title('k-means Clustering Result', fontsize = 20)

scatter = ax.scatter(principalDf['principal component 1'],principalDf['principal component 2'],
                    c=kmeans[0],s=50)

scatter = ax.scatter(cent[:,0],cent[:,1],s=50,color='black',label='cent')
ax.set_title('K-Means Clustering')
plt.colorbar(scatter)

# Creating an extra column in data for storing the cluster values
data['Cluster'] = kmeanModel.labels_
data['Cluster'].sample(n=10)

```

```

#Evaluating K-means Clustering
silhouette = silhouette_score(data_norm, predicted_pso_rep)
sse = calc_sse(centroids=pso_rep.gbest_centroids, data=data_norm, labels=predicted_pso_rep)
quantization = pso_rep.gbest_score
print('Silhouette:', silhouette)
print('SSE:', sse)
print('Quantization:', quantization)

#Popularity-based Recommendation Algorithm
UserID=10.0
cluster=2
data3=data1
data3 = data3.drop(['Genres'],axis =1)
data3.isnull().sum()
data3.dropna(inplace= True )
similar_movies_id=pd.merge(data,data3,how='outer',on='MovieID')
similar_movies_id.dropna(inplace= True )
movie_rating = data2.groupby(['MovieID'], as_index=False)
average_movie_ratings = movie_rating.agg({'Rating':'mean'})
average_movie_ratings.columns=['MovieID', 'AvgMovieRating']
recommend2=pd.merge(recommend2,average_movie_ratings,how='outer',on='MovieID')
#Identifying movies which have been rated by the current user
user_mv=recommend2.iloc[list(np.where(User_label==2))[0]]
#Identifying movies which have not been rated by the current user
user_new_movies=user_mv[user_mv['UserID']!=UserID]
#selecting the unique movies from user_new_movies
unique_movies= user_new_movies.drop_duplicates('MovieID')
#creating dataframe for displaying the list of movies
movielist = pd.DataFrame(unique_movies, columns = ['Title','AvgMovieRating'])
movielist.sort_values('AvgMovieRating', ascending=False, inplace=True)
movielist.set_index('AvgMovieRating', inplace=True)
movielist.head(20)

```

2. Appendix 2: Particle Search Algorithm

```
import numpy as np
from kmeans import KMeans, calc_sse
def quantization_error(centroids: np.ndarray, labels: np.ndarray, data: np.ndarray) -> float:
    error = 0.0
    for i, c in enumerate(centroids):
        idx = np.where(labels == i)
        dist = np.linalg.norm(data[idx] - c)
        dist /= len(idx)
        error += dist
    error /= len(centroids)
    return error
class Particle:
    def __init__(self,
        n_cluster: int,
        data: np.ndarray,
        use_kmeans: bool = False,
        w: float = 0.9,
        c1: float = 0.5,
        c2: float = 0.3):
        index = np.random.choice(list(range(len(data))), n_cluster)
        self.centroids = data[index].copy()
        if use_kmeans:
            kmeans = KMeans(n_cluster=n_cluster, init_pp=False)
            kmeans.fit(data)
            self.centroids = kmeans.centroid.copy()
```

```

self.best_position = self.centroids.copy()
self.best_score = quantization_error(self.centroids, self._predict(data), data)
self.best_sse = calc_sse(self.centroids, self._predict(data), data)
self.velocity = np.zeros_like(self.centroids)
self._w = w
self._c1 = c1
self._c2 = c2
def update(self, gbest_position: np.ndarray, data: np.ndarray):
    self._update_velocity(gbest_position)
    self._update_centroids(data)
def _update_velocity(self, gbest_position: np.ndarray):
    v_old = self._w * self.velocity
    cognitive_component = self._c1 * np.random.random() * (self.best_position - self.centroids)
    social_component = self._c2 * np.random.random() * (gbest_position - self.centroids)
    self.velocity = v_old + cognitive_component + social_component
def _update_centroids(self, data: np.ndarray):
    self.centroids = self.centroids + self.velocity
    new_score = quantization_error(self.centroids, self._predict(data), data)
    sse = calc_sse(self.centroids, self._predict(data), data)
    self.best_sse = min(sse, self.best_sse)
    if new_score < self.best_score:
        self.best_score = new_score
        self.best_position = self.centroids.copy()
def _predict(self, data: np.ndarray) -> np.ndarray:
    distance = self._calc_distance(data)
    cluster = self._assign_cluster(distance)
    return cluster

def _calc_distance(self, data: np.ndarray) -> np.ndarray:
    distances = []
    for c in self.centroids:
        distance = np.sum((data - c) * (data - c), axis=1)
        distances.append(distance)
    distances = np.array(distances)
    distances = np.transpose(distances)
    return distances
def _assign_cluster(self, distance: np.ndarray) -> np.ndarray:
    cluster = np.argmin(distance, axis=1)
    return cluster
if __name__ == "__main__":
    pass

```

3. Appendix 3: Particle Swarm Optimization Algorithm (PSO)

```
import numpy as np
from particle import Particle
class ParticleSwarmOptimizedClustering:
    def __init__(self,
                  n_cluster: int,
                  n_particles: int,
                  data: np.ndarray,
                  hybrid: bool = True,
                  max_iter: int = 100,
                  print_debug: int = 10):
        self.n_cluster = n_cluster
        self.n_particles = n_particles
        self.data = data
        self.max_iter = max_iter
        self.particles = []
        self.hybrid = hybrid
        self.print_debug = print_debug
        self.gbest_score = np.inf
        self.gbest_centroids = None
        self.gbest_sse = np.inf
        self._init_particles()

    def _init_particles(self):
        for i in range(self.n_particles):
            particle = None
            if i == 0 and self.hybrid:
                particle = Particle(self.n_cluster, self.data, use_kmeans=True)
            else:
                particle = Particle(self.n_cluster, self.data, use_kmeans=False)
            if particle.best_score < self.gbest_score:
                self.gbest_centroids = particle.centroids.copy()
                self.gbest_score = particle.best_score
            self.particles.append(particle)
            self.gbest_sse = min(particle.best_sse, self.gbest_sse)

    def run(self):
        print('Initial global best score', self.gbest_score)
        history = []
        for i in range(self.max_iter):
            for particle in self.particles:
                particle.update(self.gbest_centroids, self.data)
                # print(i, particle.best_score, self.gbest_score)
            for particle in self.particles:
                if particle.best_score < self.gbest_score:
                    self.gbest_centroids = particle.centroids.copy()
                    self.gbest_score = particle.best_score
            history.append(self.gbest_score)
            if i % self.print_debug == 0:
                print('Iteration {:04d}/{:04d} current gbest score {:.18f}'.format(
                    i + 1, self.max_iter, self.gbest_score))
        print('Finish with gbest score {:.18f}'.format(self.gbest_score))
        return history

if __name__ == "__main__":
    pass
```