Travel Review Ratings

Rajaraman Ganesan
Master Engineering
Electrical & Computer Dept.
Western University, London
rganesa@uwo.ca

Vatsal Shah
Master Engineering
Electrical & Computer Dept.
Western University, London
vshah56@uwo.ca

***Abstract*-- Advancement in technology has fundamentally changed how information produced and consumed by all users involved in travel. Traveller can now access different sources of information, and they can generate their content and share their views and experiences. Reviews shared through online has become a vital information source that affects travel in terms of both reputation and performance. However, the volume of data on the Internet has reached a level that makes manual processing almost impossible, demanding new analytical approaches — clustering technique considered as a possible solution to limit the volume of data. In travel review, based on users rating on social media say After pre-processing the dataset, the problem reduced to Clustering problem. Here, we make use of clustering algorithm techniques like the k-mean, k-medoids and CLARA algorithm using Python and R. We conclude by evaluating results and compared with custom models and available libraries in python such as Sklearn. Also, we assessed the results of the algorithm in R language with available libraries and visualized the results. Moreover, explored about Amazon SageMaker and deployed python code.**

Keywords*-- *Clustering models, k-means, k-medoids, Clara, Cluster evaluation, tourism, Sklearn, python, R, silhouette plot, Amazon SageMaker

I. INTRODUCTION

In the age of e-commerce, every industry is involved in online sales, and the hospitality and tourism industry is no exception. The participatory nature of the Internet in recent years has led to an explosive growth of travel-related user-generated content. Travel planning has become one of the significant commercial use. Sharing on the web has become an essential tool in expressing customer thoughts about a product or Service. Many tourists look for some places like fun malls, restaurants or vacation spots online in recent times [1]. After consumption customers give feedback/rating, online so online reviews have become increasingly important. They are fast, updated and available everywhere and have become the word-of-mouth of the digital age. Thus, online review plays a critical role in the tourism industry, which mainly offers services and focuses on customer satisfaction. So, people spend time online reading the review/rating backing their decision-making.

In this study, user ratings captured from Google reviews across the Europe region, and average rating ranges from 1 to 5. The dataset contains information on 25 variables, obtained from the UCI Machine Learning Repository. With these reviews, we can make the right decision about the places about to visit, nature of the user.

A. Background & Motivation

The reason for choosing the topic is to find the best places the people can visit. We process the data provided analyze and clusters the range of rating provided by the tourist. The aim is to resolve this problem by building and comparing various techniques using unsupervised learning algorithm. Moreover, to encounter the difference in the process and issues by applying the custom model and pre-built libraries.

B. Aim & Objective

The objective of the problem is to cluster the range of ratings provided by various consumers in various places they have visited. This project helps to solve problems using multiple clustering techniques such as K-means, K-medoids and CLARA algorithm and find the average ratings given by users on different places. We implement by applying various methods, custom model and pre-built libraries in python by understanding the process, compare and evaluate results. In the end, export more about cloud technology in Machine learning like Amazon SageMaker.

The steps followed to manage these goals:

1. Understand the selected dataset
2. Display some graphical information and visualize the features
3. Data pre-processing
4. Apply clustering algorithms on the dataset chosen using a custom model of k-means in Python
5. Apply k-means algorithm using pre-built libraries in python
6. Apply algorithms like k-means, k-medoids, Clara in R
7. Evaluate the model in python & R
8. Compare the model and find the optimal one
9. Explore cloud technology in Machine Learning on Amazon SageMaker

C. High-Level Overview

The primary purpose of this is to observe the ratings provided by users in various places. Clustering algorithms implemented to analyze this model. Many methods of clustering including soft clustering been used to develop this model of users review.

The paper organized as follows. Section 2 summarizes the basic properties of applied models; section 3 explores the methodology with data preprocessing. Section 4 comprises the evaluation process, and section 5 presents a summary.

II. BACKGROUND & LITERATURE SURVEY

The research work of [2] presents a comparison of three different datasets gathered from travel and tourism domain. The first dataset has 249 user records with six attributes, seconds dataset has 980 user records with ten attributes, and the third dataset has 5456 user records with 24 rating attributes. They applied various clustering techniques such as k-means, k-medoids, and CLARA and Fuzzy c-means using R packages. In the end, they concluded that the k-means algorithm performed better than other clustering algorithms.

There is much research on review rating because every people believe the rating provided online and plan his or her visit to those selected places. Many statistical methods have been applied to develop a travel review rating model, using, K-means, soft clustering algorithms.

A. Clustering:

Clustering is a set of observations into subset in the same cluster are similar in some sense. In the world of machine learning, it is an unsupervised approach. Unsupervised learning applied while there is input data, but there are no corresponding output variables associated with it. The objective of this algorithm is to find different groups within elements in the dataset. Moreover, it finds the structure in the data so that elements of the same cluster are more like each other than to those from different clusters. It has various uses in many fields such as machine learning, pattern recognition, image analysis, information retrieval, bio-informatics, data compression, and computer graphics [2].

B. K-means Algorithm:

K-means is one of the simplest algorithms uses unsupervised learning method to solve clustering issues. It mainly used when having unlabeled data. This algorithm goal is to find groups in data, with the number of groups represented by the variable.

$$WCV(C_k) = \sum_{x_i \in \mu_k} (x_i - \mu_k)^2 \qquad [3]$$

Where,
x_i = entity belong to the cluster C_k
C_k & μ_k = mean of all entities forming the cluster.

The following steps followed to implement the k-means algorithm:

1. Determine the total no: of clusters to be formed.
2. Start by identifying centroids initially and random sampling done within the dataset to find initial points.
3. Calculate the distance between the entity and each of centroids and assign the entity to the cluster close to the centroids.
4. Revise the cluster centroid by calculating mean values of all entities. Repeat this for each cluster.
5. Reduce the total within-cluster variation. Also, repeat the above step 3 and four until the max number of iterations reached.

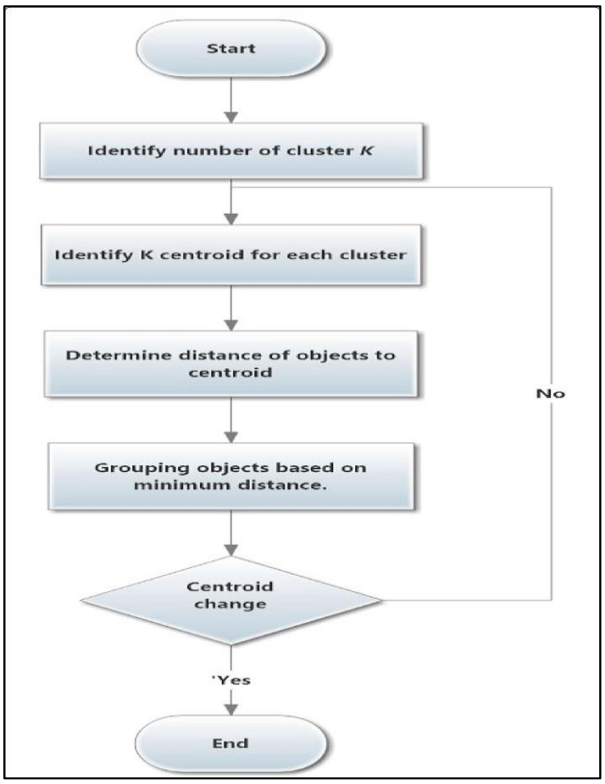


Fig (2.1) Process for the k-means algorithm

C. K-medoids Algorithm:

The k-medoids algorithm works similar to the k-means algorithm, but less sensitive to the presence of noises and outliers in the data set. The critical difference is in fitting the centroids. While in the k-means algorithm the centroids need not be the actual entities from the dataset, k-medoids algorithm chose an entity from the dataset whose average dissimilarity to all the other entities is minimum as the medoid.

The steps to perform the k-medoids algorithm given below:

1. Determine k, the total number of clusters to be populated
2. Identify k initial medoids to start. Medoids always selected from the members of the dataset.

3. For each entity in the dataset
a. Calculate the distance between the entity and each of the k medoids
b. Allocate the entity to a cluster whose medoid is the nearest
4. For each of the k clusters
a. Scan for an entity that can reduce the average dissimilarity within the cluster.
b. If an entity found that can reduce the objective function, swap the medoid and the entity in consideration.
5. If there is any change to at least one medoid in step 4, go to step 3 else end.

D. CLARA Algorithm:

CLARA (Clustering for Large Applications) algorithm formulated as an extension to a k-medoids algorithm to deal with datasets containing a vast number of entities (when it ranges in several thousand). To overcome the scalability challenges of the k-medoids algorithm, it needs high computing time and large memory requirements. It achieved by utilizing the sampling approach.

The below given are the steps to perform Clara algorithm:

1. Split the dataset randomly into multiple subsets Having a fixed size.
2. Apply the PAM algorithm on each subset and choose the Corresponding k medoids.
3. For each entity in the complete dataset
a. Calculate the distance between the entity and each Of the k medoids.
b. Allocate the entity to a cluster whose medoid is the Nearest.
4. For each subset
a. Calculate the average dissimilarities of the entities to their closest medoid (quality or goodness of the Clustering).
5. Select sub-dataset with the minimum average dissimilarity.

III. METHODOLOGY

In methodology, data description, independent variable, and dependent variable described with the scale of variables. Moreover, in the process data preprocessing and feature engineering described below.

A. Data Description & Preparation:

This dataset consists of 5456 total instances and 25 features including:

TABLE I
ATTRIBUTES OF THE DATASET

Attribute No	Description
1	Unique user ID
2	Average ratings on churches
3	Average ratings on resorts
4	Average ratings on beaches
5	Average ratings on parks
6	Average ratings on theatres
7	Average ratings on museums
8	Average ratings on malls
9	Average ratings on zoo
10	Average ratings on restaurants
11	Average ratings on pubs/bars
12	Average ratings on local services
13	Average ratings on burger/pizza shops
14	Average ratings on hotels/other lodgings
15	Average ratings on juice bars
16	Average ratings on art galleries
17	Average ratings on dance clubs
18	Average ratings on swimming pools
19	Average ratings on gyms
20	Average ratings on bakeries
21	Average ratings on beauty & spas
22	Average ratings on cafes
23	Average ratings on viewpoints
24	Average ratings on monuments
25	Average ratings on gardens

There are 24 categories reviewed by users. It shows in the below rating ranger from 1 to 5. Each category has a various number of users rating. For a better understanding of the data, visualization with hist and plot diagram as below:

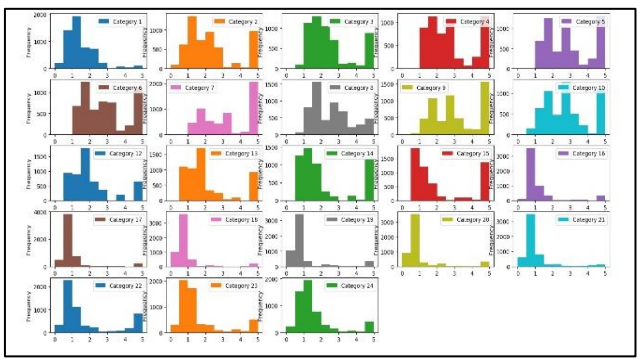


Fig (3.1) Visualization of data with hist diagram

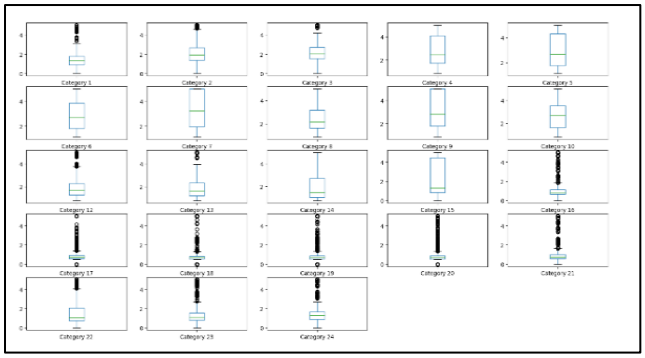


Fig (3.2) Visualization of data with box-plot diagram

In this dataset, to understand the importance of features with compare to each other and find the best features to evaluate the model used heatmap as shown below:

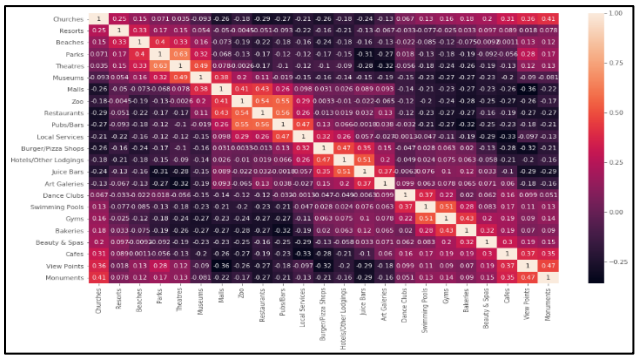


Fig (3.3) Visualization of data with heatmap diagram

B. *Data Cleaning & Normalizing:*

Mostly, data gathered from various sources, which have missing values and noises. Because of this, data cleaning is a crucial step to perform before applying an algorithm. There are various approaches for data cleaning. In Python, isna() used to identify null values of all attributes and isna().sum() gives the summery of all attributes with the number of null records. It replaced with the mean value. In python, it implemented as fillna() function which replaces all missing values with the mean value.

Scaling or normalizing of the attribute is a practice [2] can perform in clustering problem. By performing normalization all, the attributes are in the same range.

IV. EVALUATION

The objective of a project is to identify the optimal number of clusters along with the best approach to apply k-means algorithms. The objective of a clustering algorithm is to accomplish minimum inter-cluster similarity among clusters and maximum intra-cluster similarity within each cluster [4][5].

A. *Elbow Method:*

A method of interpretation and validate of consistency within cluster analysis designed to help to find the appropriate number of clusters in the dataset [6][7].

The optimal number of clusters defined as follow:

1. Compute clustering algorithm (e.g., k-means clustering) for different values of k. For instance, by varying k from 1 to 10 clusters.
2. For each k, calculate the total within-cluster sum of square
3. Plot the curve of was according to the number of clusters k.
4. The point of a bend in the below plot considered as the appropriate number of clusters

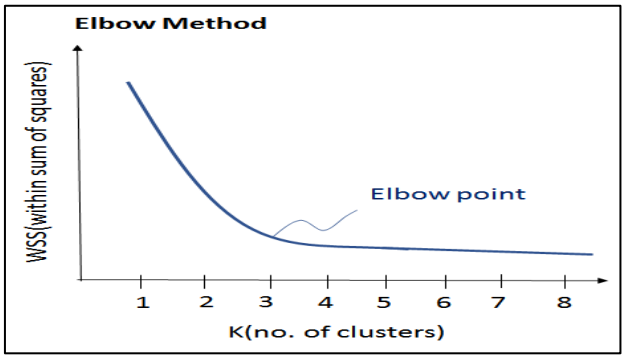


Fig (4.1) Elbow method graph represents WSS vs No. of clusters

Silhouette Plot:

Silhouette is a method of interpreting and validate consistency within clusters of data. The technique provides a graphical representation of how each object has classified. It displays a measure of how close each point in one cluster is to points in the neighbouring clusters and thus provides a way to assess parameters like several clusters visually. The silhouette calculated with any distance metric, such as the Euclidean distance or the Manhattan distance. [8]

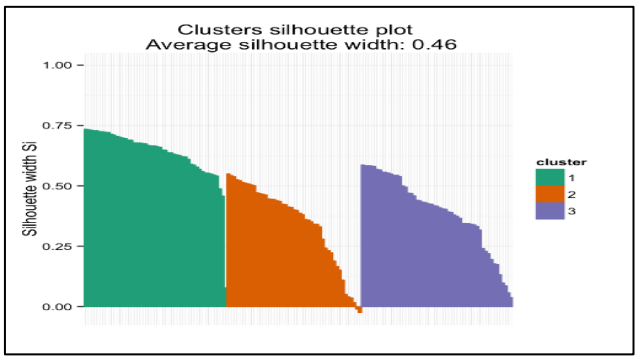


Fig (4.2) Represents the Silhouette plot

B. *Euclidean distance:*

The distance between two points in either the plane or 3D space measures the length of a segment connecting the two points. The way of plotting the distance between two points.

$$d_{euc}(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad [9]$$

Where x and y are two vectors of length n.

Kmeans () function returns a list as below:

- cluster: each point allocated to a cluster

- centers: cluster means
- totss: The total sum of squares and measures the total variance in the data.
- withinss: Vector of the within-cluster sum of squares, one component per cluster
- tot.withinss: Total within-cluster sum of squares,
- betweenss: The between-cluster sum of squares
- size: Number of observations in each cluster

```
K-means clustering with 3 clusters of sizes 1093, 2460, 1903

Cluster means:
  Churches  Resorts  Beaches    Parks Theatres  Museums    Malls      Zoo Restaurants Pu
bs/Bars
1 2.246844 2.729259 2.584986 2.379790 2.099021 1.917859 2.003001 1.580366    1.909735  1
.615938
2 1.092085 1.940724 1.916549 2.100972 2.257573 2.710484 3.919435 3.081898    4.019260  3
.572687
3 1.471403 2.574388 3.174824 3.936054 4.359496 3.690420 3.391550 2.392943    2.669911  2
.575060
  Local Services Burger/Pizza Shops Hotels/Other Lodgings Juice Bars Art Galeries Dance
clubs
1       1.548609           1.414959              1.502031   1.831189     2.336203    1.6
71537
2       3.180236           2.491467              2.579427   2.923260     2.868951    1.0
59378
3       2.310368           1.925265              1.896837   1.450673     1.275864    1.0
90310
  Swimming Pools      Gyms  Bakeries Beauty & Spas     Cafes View Points Monuments  Gard
ens
1      1.7643367 1.7999451 2.2789113     2.1559469 1.8064867    2.7278957 2.3618207 2.451
043
2      0.7661585 0.6089959 0.7034512     0.7320488 0.6968130    0.9943862 0.9045122 1.059
728
3      0.7176458 0.5368471 0.5622438     0.6826590 0.8307725    2.1666579 1.8649711 1.696
816
```

```
Within cluster sum of squares by cluster:
[1] 34299.27 74583.77 56573.43
 (between_SS / total_SS =  23.3 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweens
s"
[7] "size"         "iter"         "ifault"
```

Fig(4.3) Represents K-means function return list

C. Results in R

The graphical representation is shown below to have a better understanding of the algorithm we implemented using the R language. Clustering is performed using k-means, k-medoids (portioning around medoids), and clustering for large applications (CLARA), approaches and the resultant clusters are plotted using fviz_cluster() function.

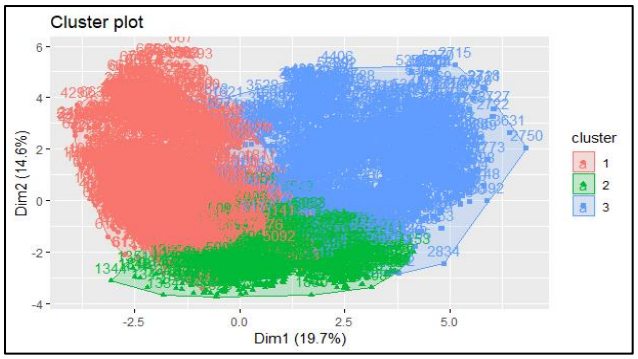
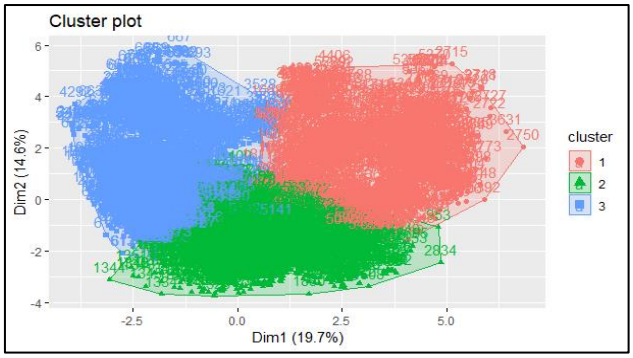


Fig (4.4) Cluster plot using the k-means algorithm



Fig(4.5) Cluster plot using the k-medoids algorithm

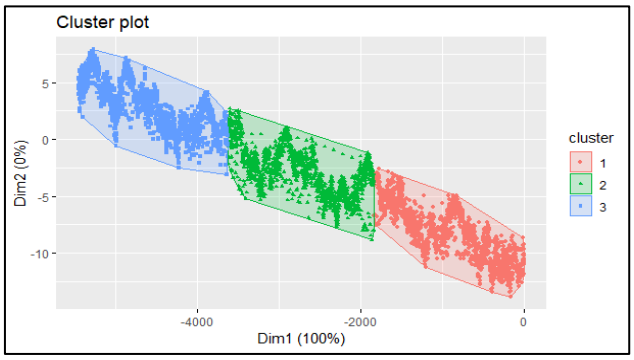


Fig (4.6) Cluster plot using CLARA algorithm

Silhouette Plot technique provides a graphical representation of how well each object has classified. Each vertical line corresponds to an element.

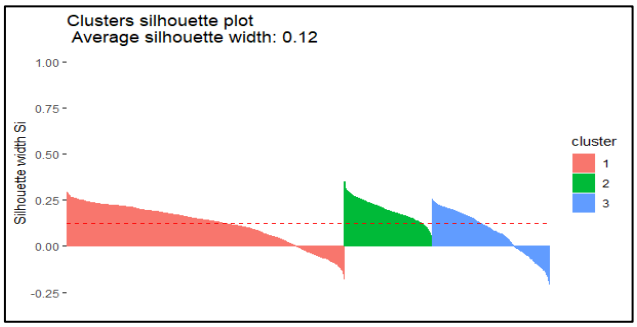


Fig (4.7) Silhouette Plot for various clusters

	cluster	size	ave.sil.width
1	1	3140	0.12
2	2	992	0.19
3	3	1324	0.08

Fig(4.8) Represents Average Silhouette width

The below given is the elbow plot is a method to validate an optimal number of cluster. Also, help to find the appropriate number of clusters in a dataset.

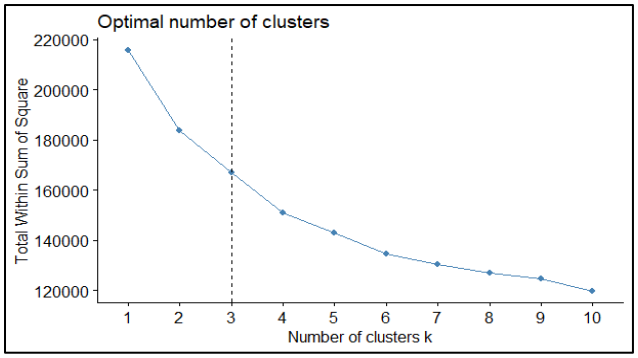


Fig (4.9) Optimal no: of the cluster for elbow plot in R

D. Results in Python

When implemented the code with python programming language we have a graphical representation of plotting of centroid given below. It is applied random centroid to the data and using the Euclidean distance calculate and assign each value to its closest cluster. In the process, every time we find the distance from the cluster and assign to the nearest one. Moreover, this process continues until error becomes zero and no more changes in centroid.

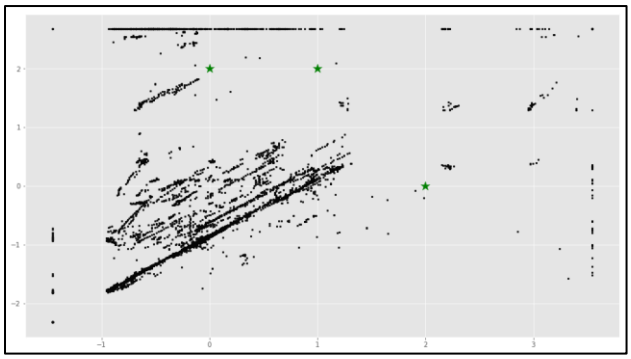


Fig (4.10) Represents plotting of centroid randomly

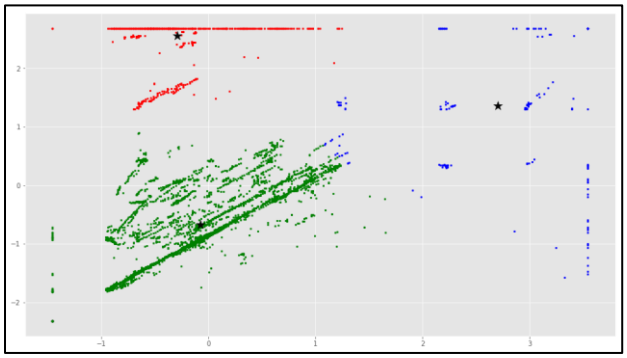


Fig (4.11) Represents the assignment of the new centroid

After creating a separation of data in 3 various clusters, the below graph represents data in the 3D frame. It is showing the separation of data in 3 different clusters. The centroid represented as a start in the center of the data and data points shows in red, green and blue colour.

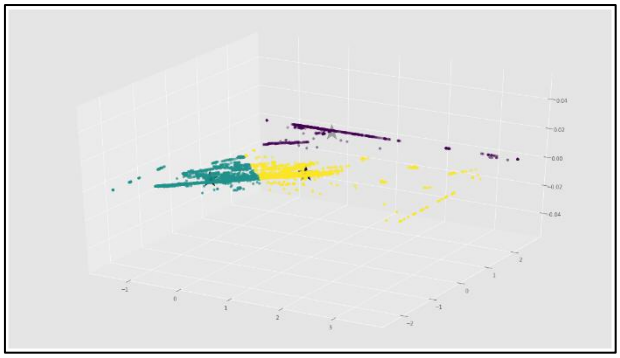


Fig (4.12) Represents clustering in 3D

Elbow graph representation in python and option value for K is 3.

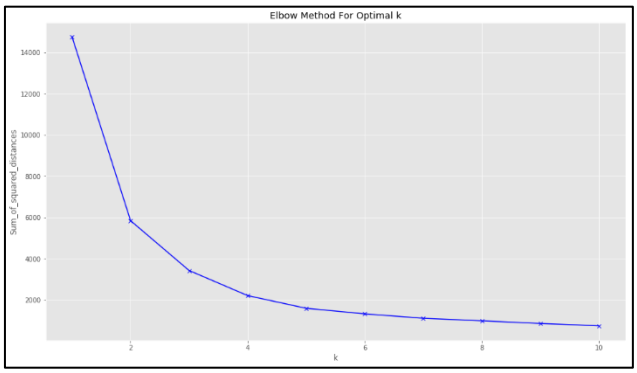


Fig (4.13) Optimal no: of the cluster for elbow plot in Python

E. Time-complexity:

Time-complexity in python and R includes time for fitting an algorithm, predicting an algorithm, total time to run the code.

TABLE II
TIME-COMPLEXITY IN PYTHON

In Python	Time (sec)
Time for the custom model - Assigning each value to its closest cluster	1.728 s
Time for fitting a model	0.102 s
Time for predicting a model	0.001 s
Total time to run a code	16.59 s

TABLE III
TIME-COMPLEXITY IN R

In R	Time (sec)
Time for k-means	0.84 s
Time for k-medoids	8.50 s
Time for CLARA	14.2 s
Total time to run a code	23.9 s

F. Tools & Libraries used for implementation:

Python: To implement the code we have made use of Python 3.6.1 version. We had made use of numpy library for multidimensional array used to store the same datatype. Pandas library provides high performance and used for data analysis tools. We make use of matplotlib.pyplot for comprehensive 2D/3D plotting and displaying understandably. Seaborn is a visualization library based on matplotlib.

R: The code implemented using 3.5.2 version. We made use of cluster library to find data in groups, ggplot2 to create decorative visualization and factoextra library to visualize the clusters.

V. SUMMARY & FUTURE WORK

We studied the data, checking for data unbalancing, preparation, visualizing features and understanding relationships between various elements. We are building the model with K-means, K-medoids and CLARA algorithm from R libraries, K-means algorithm with python libraries and custom model. Then select the best cluster value k, compared and checked with elbow graph. Finally, conclude the results and ended with the best one. Moreover, we worked on Amazon SageMaker to run python code and explore scope in cloud technology for Machine learning.

In the future work, we would like to explore more on unsupervised learning approach in Machine learning and select unlabeled dataset of Gigabyte, save on Amazon S3, run a model on Amazon Sage Maker, save and use a model for real-time data.

VI. REFERENCES

[1]. Celebi, M. E., Kingravi, H. A., and Vela, P. A. (2013). "A comparative study of efficient initialization methods for the k-means clustering algorithm." Expert Systems with Applications.

[2]. Renjith, Shini, and C. Anjali. "A personalized mobile travel recommender system using the hybrid algorithm." In Computational Systems and Communications (ICCSC), 2014 First International Conference on, pp. 12-17. IEEE, 2014

[3]. Tryon, Robert C. (1939). Cluster Analysis: Correlation Profile and Orthometric (factor) Analysis for the Isolation of Unities in Mind and Personality. Edwards Brothers.

[4]. Maulik, Ujjwal, and Sanghamitra Bandyopadhyay. "Performance evaluation of some clustering algorithms and validity indices." IEEE Transactions on Pattern Analysis and Machine Intelligence 24, no. 12 (2002): 1650-1654.

[5]. Kovács, Ferenc, Csaba Legány, and Attila Babos. "Cluster validity measurement techniques." In a 6th International symposium of hungarian researchers on computational intelligence. 2005.

[6]. David J. Ketchen, Jr; Christopher L. Shook (1996). "The application of cluster analysis in Strategic Management Research: An analysis and critique." Strategic Management Journal.

[7]. Trupti M. Kodinariya, Dr. Prashant R. Makwana. "Review on determining the number of Cluster in K-Means Clustering." In International Journal of Advance Research in Computer Science and Management Studies. Volume 1, Issue 6, November 2013

[8]. Selecting the number of clusters with silhouette analysis on KMeans clustering, Sklearn (Link)

[9]. Partitioning cluster analysis: Quick start guide - Unsupervised Machine Learning (Link)

CONTRIBUTIONS

Rajaraman Ganesan: He worked on data preparation and understanding of clustering algorithms. He applies algorithm like k-means, k-medoids and CLARA in R. His role is to find the best value of cluster, compare and check with elbow graph in R. At the end, evaluate the best results after applying libraries. He also found the time-complexity to run each algorithm and total time to run a code in R.

Vatsal Shah: He worked on dataset understanding, data pre-processing. His role is to understand and apply k-means algorithms with custom models. It includes create equations, find distance and select the best value of cluster K number and evaluate results. He created the output.txt file to write all results of the process in python. Moreover, found the time-complexity for k-means in fitting and predicting a mode, total time to run the code. Lastly, he used Amazon SageMaker to explore cloud technology in Machine learning and run a python code to find the results.