Journal of Applied Informatics and Computing (JAIC)
Vol.8, No.1, July 2024, pp. 221~225
e-ISSN: 2548-6861

Visit Recommendation Model: Recursive K-Means Clustering Analysis of Retail Sales Data

Bagus Kristomoyo Kristanto [1]*, Syntia Widyayuningtias Putri Listio[2], Mukhlis Amien[3]**, Panji Iman Baskoro[4]****
* Sistem Informasi , Sekolah Tinggi Informatika & Komputer Indonesia
** Informatika, Sekolah Tinggi Informatika & Komputer Indonesia
bagus.kristanto@stki.ac.id [1], syntia@stiki.ac.id [2], amien@stiki.ac.id [3] panjidia995@gmail.com[4]

Article Info

Article history:

Received 2024-07-03
Revised 2024-07-15
Accepted 2024-07-16

Keyword:

Cluster Analysis,
K-Mean Clustering,
Retail Sales Data Clustering,
Visit Recommendation.

ABSTRACT

In the context of retail distribution, this study employs recursive K-means clustering on retail sales data to optimize clusters of nearest-distance stores for salesperson route recommendations. This approach addresses the stochastic salesperson problem by generating effective routes, enhancing cost reduction, and improving service efficiency. The recursive K-means algorithm dynamically adjusts to continuous changes in store numbers, locations, and transaction data. Consequently, this research successfully developed a model that automatically re-clusters the data with each change, providing continuously updated and effective store recommendations.

I. INTRODUCTION

In the context of retail business distribution systems, it is crucial to put emphasis on cost reduction and service enhancements. Technology can accomplish this by offering a comprehensive system that streamlines the management and monitoring of all product delivery processes by salespersons to individual retail stores. The primary objective of the salesperson monitoring and management system is to gather data and information related to the sales progress and performance of each salesperson[3]. This aspect holds significant importance for retail businesses or companies [1][2]. However, there is still an issue with the stochastic salesperson problem when considering distance measured by travel time and tour due dates. In the initial studies of the stochastic routing problem, the objective of the model was to minimize the time it takes to complete all the routes or to maximize the probability of finishing them by a particular time limit. [12][13]. Given the previously mentioned problem, it is crucial to create a system that can offer suggestions for the most effective route. [5][6].

Through employing the data from the sales monitoring system, including transaction data and geographical information for each visited store, the system can generate effective route recommendations and enhance the company's marketing strategy[4]. Research in the field of grouping position points to determine an effective route often applies k-means clustering to create optimal clusters and find the shortest route [7][8][9]. In order to obtain the effective k cluster with the shortest distance, combine with the recursive technique that can estimate the cluster center by recursively checking the cluster size for each iteration [7][10][11]. The main goal of this study is to improve the clustering of store points using retail sales data, which will help the system provide better route recommendations for traveling salespeople. This will be achieved by applying recursive K-means clustering.

II. METHOD

A. Study Area and Data sets

The data analyzed in this study consists of store visits and transactions made by salespersons during a three-month period from April 2024 to June 2024. Using a dataset of GPS pings, we analyzed the movements of 500 salespersons across Indonesia to derive the store locations. This dataset contains accurate longitude and latitude coordinates for every store, comprehensive earnings statistics for each salesperson, and detailed timelines of each salesperson's trips to stores within their assigned areas.

B. Cluster Analysis: K-means clustering.

Cluster analysis is a dimension reduction approach that aims to improve comprehension of complex data structures by identifying patterns. The K-means algorithm remains popular in cluster analysis due to its rapidity, effectiveness, and straight forwardness [14]. The K-means clustering technique typically divides each object or observation into a predetermined number of clusters, denoted as *(k)*[15][16]. However, in this case, the number of new stores is continually updated, making it challenging to determine the number of clusters for each area. The only predefined constraint is the maximum number of stores assigned to each salesperson. In response to the above problem, we employed recursive clustering using K-means clustering technique to group retail data.

C. Recursive K-Means Clustering

For the purpose of achieving our research objective of clustering retail data to offer visit route suggestions, we utilize a sequential clustering technique with K-means. This method is adept at handling an unspecified number of clusters and produces a new cluster of stores for each area.

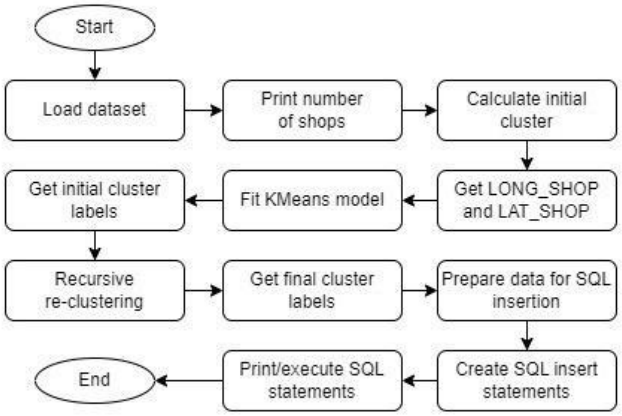

Figure 1. research step flowchart

The main step for this clustering method is the recursive re-clustering step. This step involves a series of actions and steps to process the data, such as running k-mean clustering, and then repeating the calculation until each cluster has a variable that is less than the maximum value for a shop within it (Figure 2).

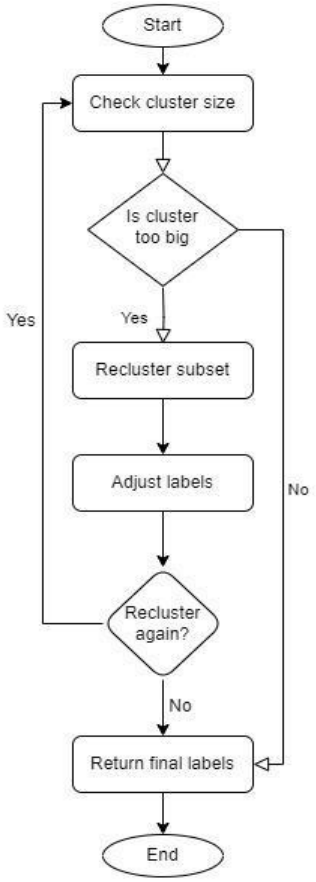

Figure 2 Recursive Process. K-Means Clustering in Algorithm

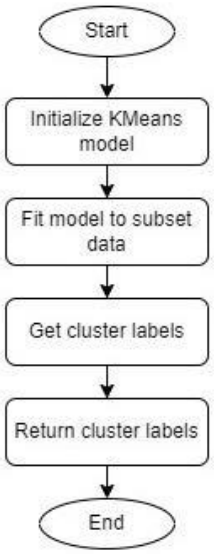

Figure 3. K-Means Clustering

III. Result and Discussion

A. Preparation Dataset

For the implementation, we have gathered a total of 3951 samples of store data from the salesperson management and monitoring systems. It is important to note that we specifically focused on sampling store locations in the surrounding area of Sidoarjo, East Java. We extract data from the database and clean several stores that haven't had any transactions in the past 3 months since April 2024. We used this preprocessing data to query the database and export the sample data to a CSV file. In this research, we need to declare the maximum number of stores in one cluster. In k-means clustering and reclustering, we use this variable until each cluster has less than the number we set.

B. Recursive K-Means Clustering

In this research, the primary requirement for recursive k-means clustering comes from the constraints imposed by the need for each cluster to have a subset (in this case, a store) that is either less than or equal to the number defined in the previous step. Each cluster that has more than the maximum number will recalculate. The algorithm will divide this cluster into two or more smaller clusters, each containing a new centroid. The recursive algorithm will recalculate and create a new cluster using the K-means clustering model. The algorithm will repeat this process until all clusters have less than or equal to the maximum number of stores defined in the previous step.

C. Code Recursive K-Means Clustering

```
# Define recursive re-clustering function
def recluster_until_fit(X, labels, jumlah_toko_perhari, start_label):
    unique_labels = np.unique(labels)
    final_labels = labels.copy()
    new_label_start = start_label

    for cluster_id in unique_labels:
        cluster_indices = np.where(labels == cluster_id)[0]
        cluster_size = len(cluster_indices)

        if cluster_size > jumlah_toko_perhari:
            new_cluster_count = ceil(cluster_size / jumlah_toko_perhari)
            new_labels = recluster_subset(X[cluster_indices], new_cluster_count)

            adjusted_labels = new_labels + new_label_start
            final_labels[cluster_indices] = adjusted_labels

            new_label_start += new_cluster_count

    counts = pd.Series(final_labels).value_counts()
    if any(counts > jumlah_toko_perhari):
        return recluster_until_fit(X, final_labels, jumlah_toko_perhari, new_label_sta
    else:
        return final_labels

# Initialize start label for re-clustering
new_label_start = max(labels) + 1

# Perform recursive re-clustering
final_labels = recluster_until_fit(X, labels, jumlah_toko_perhari, new_label_start)

# Update labels with the final labels
labels = final_labels
```

Figure 4. Recursive K-Means Clustering and Re-clustering

This image illustrates the primary algorithm we develop in our research. We write this code in the Python programming language. Python is a low-level and well-known programming language for clustering models. As previously explained, we will repeat the clustering process until all clusters contain less or equal to the maximum number of stores, which takes an average of 3.5 seconds to calculate 3951 samples. Once the algorithm completes the calculations, including reclustering, it assigns a unique ID to each cluster. Creating a unique cluster is used as a visiting recommendation model for salespeople within the system.

D. The code aims to illustrate clustering on a map.

```
# Visualization with Folium
map_center = [np.mean(X[:, 1]), np.mean(X[:, 0])]
mymap = folium.Map(location=map_center, zoom_start=10)

# Define a list of colors for clusters
colors = ['blue', 'green', 'red', 'orange', 'purple', 'pink', 'black', 'gray', 'brown',
'darkblue', 'darkgreen', 'darkred', 'darkpurple', 'darkorange', 'lightblue', 'lightgreen',
'lightred', 'lightgray', 'lightyellow', 'cadetblue', 'beige', 'darkgray', 'darkcyan', 'lightcyan']

# Add markers for each shop location with cluster color
for i in range(len(X)):
    cluster_color = colors[labels[i] % len(colors)]  # Cycle through colors for each cluster
    folium.CircleMarker(location=[X[i, 1], X[i, 0]], radius=2, color=cluster_color).add_to(mymap)

# Display the map
mymap.save("rute.html")
```

Figure 5. The code aims to visualize the cluster in a map.

Figure 5 shows how we draw the cluster onto the map using longitude and latitude for each store. This step helps us visualize our clustering from the previous step and allows us to see how the cluster integrates with the map. It also aids in ensuring that we group each cluster using the same color. We use the same colour to identify a cluster. It has transformed into a recommendation for a salesperson to visit. At this point, we only draw a visiting recommendation for the salesperson, not a route to visit the store in a cluster.

E. Result for K-Mean Clustering and Visit Recommendation

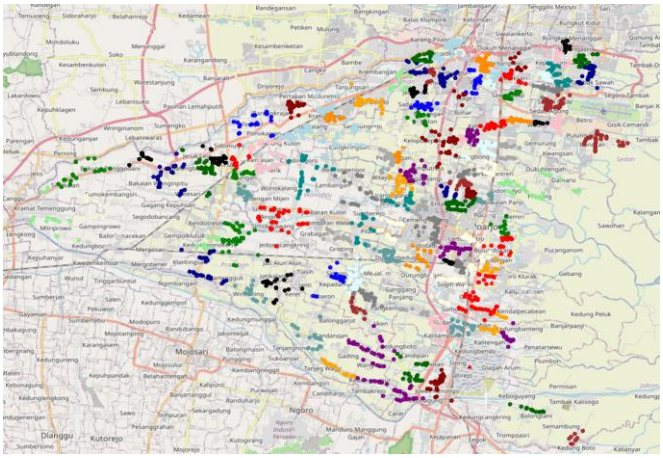

Figure 6. K-Means Clustering

Figure 5 show utilize data visualization to represent the mapping of all the collected store data, and employ K-means clustering to group the areas. Additionally, it helps to ensure that we categorize each cluster using the identical color. A cluster is identified with a uniform color. It has evolved into a suggestion for a salesperson to make a visit. The salesperson management system will display only the store data that is situated within a single cluster, based on this clustering outcome.

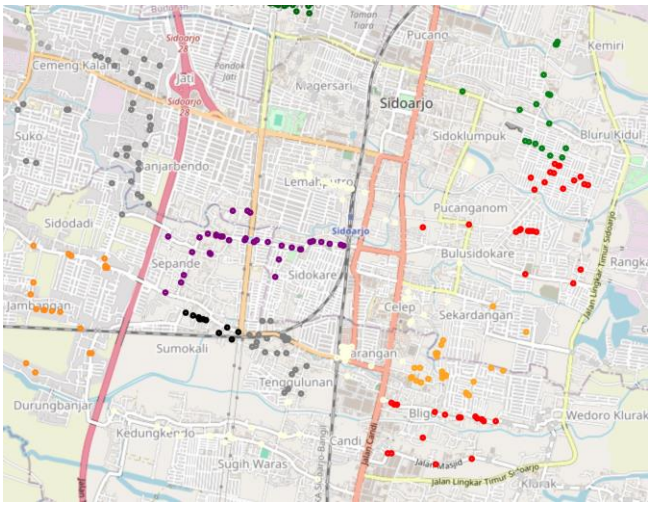

Figure 7. K-Means Clustering

Figure 7 displays an in-depth look of each cluster, with each node representing the precise GPS location of the store. This cluster has demonstrated its effectiveness in cluster data based on its distance. Additionally, it has been re-clustered whenever additional nodes are added or when the maximum number of stores in one cluster is reached. The next step involves integrating the clustering data into the sales management system, as each store belongs to a unique ID-labeled cluster.

F. Data Integration to Salesperson System Management

```
# Prepare data for SQL insertion
# Add route_group column
data['LABEL'] = labels
data['ID_USER'] = 'cd333716'
data['WEEK'] = '1'
data['ID_RUTE'] = range(1, len(data) + 1)
data['route_group'] = (data.index // 40) + 1

# Create SQL insert statements
insert_statements = []
for index, row in data.iterrows():
    insert_statements.append(f"INSERT INTO md_route (ID_RUTE, ID_USER, WEEK, ID_SHOP, route_group)
VALUES ({row['ID_RUTE']}, '{row['ID_USER']}', '{row['WEEK']}', {row['ID_SHOP']}, {row['route_group']});")

# Print or execute the SQL statements
for stmt in insert_statements:
    print(stmt)

# If you need to execute the statements on an SQLite database (for demonstration purposes):
# conn = sqlite3.connect(':memory:')  # Replace with your database connection
# cur = conn.cursor()
# cur.execute('''CREATE TABLE md_route (ID_RUTE INTEGER, ID_USER TEXT, WEEK TEXT, ID_SHOP INTEGER);''')
# for stmt in insert_statements:
#     cur.execute(stmt)
# conn.commit()
# conn.close()
```

Figure 8. Code for insert clustering into salesperson system

After receiving the cluster recommendation, the final step involves inserting the cluster data into the sales management systems. This process consists of several procedures to insert cluster data into the database. The process involves preparing the data in a SQL statement, mapping the cluster data into a table in the database, and then joining this cluster data with the salesperson. We will use this cluster data to recommend visits to salespersons in Sidoarjo, East Java. We need an average of 2–3 seconds to insert 3591 sample data into sales management systems after several tries. Once we inserted the data, the administrator of the salesperson management system could view the visiting recommendations for each salesperson.

This research successfully applied clustering techniques to retail data, primarily focusing on store location mapping. The objective was to address the challenges related to store visit recommendations and clustering based on nearest distances. Furthermore, we have implemented a recursive K-means algorithm that is capable of dynamically adjusting to the continual changes in the number and position of stores.

We could still broaden this study in specific domains, like determining the shortest route for each cluster, enhancing the model with additional sample data from stores that are more scattered, and automating the scheduling of visits based on the salesperson's location.

IV. CONCLUSION

This research effectively showed the use of clustering techniques in analyzing retail data, with a specific emphasis on store location mapping and visit recommendations. Using a recursive K-means clustering algorithm, we tackled the issue of dynamically updating the number of stores, making it more difficult to determine clusters for each area. Our method guarantees that every cluster meets the predetermined limit of stores per salesperson.

The recursive K-means algorithm proved effective in dynamically adjusting to changes in the number and positions of stores, thereby providing optimized route recommendations for traveling salespeople. This method enhances the management and monitoring of product delivery processes, ultimately contributing to cost reduction and service efficiency in the retail business distribution system.

Future research could expand on these findings by exploring domains such as determining the shortest route for each cluster, incorporating additional sample data from more dispersed stores, and automating visit scheduling based on the salesperson's location. These enhancements would further improve the effectiveness and practicality of the clustering model in real-world retail scenarios.

REFERENCE

[1] Levy, M., & Sharma, A. Relationship among Measures of Retail Salesperson Performance. Journal of the Academy of Marketing Science, 21(3), 231-238, 1993..

[2] James S. Boles, Barry J. Babin, Thomas G. Brashear & Charles Brooks, An Examination of the Relationships between Retail Work Environments, Salesperson Selling Orientation-Customer Orientation and Job Performance, Journal of Marketing Theory and Practice, 9:3, 1-13, 2001.

[3] Abhijit Guha, Dhruv Grewal, Praveen K. Kopalle, Michael Haenlein, Matthew J. Schneider, Hyunseok Jung, Rida Moustafa, Dinesh R. Hegde, Gary Hawkins, How artificial intelligence will affect the future of retailing, Journal of Retailing, 97(1), 28-41, 2021.

[4] Auke Hunneman, Tammo H.A. Bijmolt, J. Paul Elhorst, Evaluating store location and department composition based on spatial heterogeneity in sales potential, Journal of Retailing and Consumer Services, 2023.

[5] Nagel, D.M., Cronin, J.J., Bourdeau, B.L., Hopkins, C.D., Brocato, D., Retailing in the Digital Age: Surviving Mobile App Failure: An Abstract. In: Rossi, P., Krey, N. (eds) Finding New Ways to Engage and Satisfy Global Customers. AMS WMC 2018. Developments in Marketing Science: Proceedings of the Academy of Marketing Science. Springer, Cham, 2019.

[6] S. Anily, A. Federgruen, One Warehouse Multiple Retailer Systems with Vehicle Routing Costs. Management Science, 36(1):92-114, 1990.
[7] Moussa, Hassan. Using Recursive KMean and Dijkstra Algorithms to Solve {CVRP}. arXiv:2102.00567
[8] Alfiyatin, A. N., Mahmudy, W. F., & Anggodo, Y. P. (2018). K-Means Clustering and Genetic Algorithm to Solve Vehicle Routing Problem with Time Windows Problem. Indonesian Journal of Electrical Engineering and Computer Science, 11(2), 462.
[9] N. P. Barbosa, E. S. Christo, and K. A. Costa, "Demand forecasting for production planning in a food company," ARPN J. Eng. Appl. Sci., vol. 10, no. 16, pp. 7137–7141, 2015.
[10] V. K. Anand, S. K. A. Rahiman, E. Ben George and A. S. Huda, "Recursive clustering technique for students' performance evaluation in programming courses," 2018 Majan International Conference (MIC), Muscat, Oman, 2018, pp. 1-5
[11] L. A. Maglaras and J. Jiang, "OCSVM model combined with K-means recursive clustering for intrusion detection in SCADA systems," 10th International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness, Rhodes, Greece, 2014, pp. 133-13
[12] Laporte, G., Louveaux, F.V., Mercure, H., 1992. The vehicle routing problem with stochastic travel time. Transp. Sci. 26 (3), 161-170
[13] Kenyon, Astrid S. and David P. Morton. "Stochastic Vehicle Routing with Random Travel Times." Transp. Sci. 37 (2003): 69-82.
[14] Chen, Angela H. L., Liang, Yun-Chia, Chang, Wan-Ju, Siauw, Hsuan-Yuan, Minanda, Vanny, RFM Model and K-Means Clustering Analysis of Transit Traveller Profiles: A Case Study, Journal of Advanced Transportation, 2022, 1108105, 14 pages, 2022. https://doi.org/10.1155/2022/1108105
[15] MacQueen J., Some methods for classification and analysis of multivariate observations, Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1, June 1967, Berkeley, CA, USA, no. 14, 281–297.
[16] Chowlur Revanna, J. K., & Al-Nakash, N. Y. B. (2022). Vehicle routing problem with time window constraint using KMeans clustering to obtain the closest customer. Global Journal of Computer Science and Technology: Neural & Artificial Intelligence, 22(1), Version 1.0.