An analytical revisit to improve tourist attractions implementing classical clustering algorithms

Roisul Islam Rumi
School of Computer Science
The University of Windsor
Ontario, Windsor
rumir@uwindsor.ca

2nd Naga Jyothirmayee Dodda
School of Computer Science
The University of Windsor
Ontario, Windsor
doddan@uwindsor.ca

3rd Sumaiya Deen Muhammad
School of Computer Science
The University of Windsor
Ontario, Windsor
deenmuh@uwindsor.ca

***Abstract*—Unsupervised learning is one of the most critical segments in Artificial Intelligence. The amount of data generated by users is so immense that it is impossible to manually label them, which increases the importance of clustering algorithms. Additionally, tourism is a big industry, and travelers use reviews and from social media and google. Many works have been done featuring the user interest group; however, we have taken this a step further by focusing on the attractions in this experiment. Our investigation prioritizes personnel involved in managing the attractions in Europe by providing suggestions on the attractions that need improvement. This experimental analysis incorporates three classical clustering algorithms intending to find out the least liked attractions by tourists.**

***Index Terms*—travel rating, unsupervised learning, clustering, kmeans++, dbscan, agglomerative clustering, data analysis.**

I. Introduction

The internet has been a radically popular source to look for travel destination information. At present, it is a very common habit of people to search tourism related information, for instance, ratings of hotels, restaurants, bars, and sometime they read user reviews of different tourist locations like beach, park, cinema, monuments, etc. Due to massive information availability on online, people make holiday plans after researching about the travel places.

For example, Fig-1 shows how frequently people searched, ‘Tourist destination in Europe’ in google from all over the world between 2016 and 2019 according to google-trend data where x-axis indicates time-frame and y-axis specifies score in a range of 0 to 100 according to search trend [1]. We can see during the the search trends reached the zenith in summer when individuals usually make plans for trips. [2] [3]

II. Problem Statement

In this study, we attempt to evaluate travellers’ ratings on different tourist domains, such as Food and beverage, accommodation, natural places, historical place and so on. By analysing the data extracted from the ratings we believe we will be able to find out which zones need attention to improve their services towards visitors.

The objective of this study is to answer the following question:

Which attractions should be prioritized more by the government and local authorities to provide better customer satisfaction in Europe?

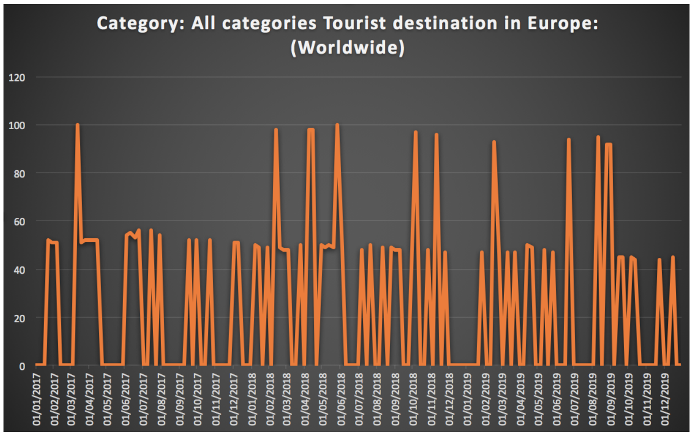


Fig. 1. Google search trend.

The paper is structured as follows. We first present the methods used in our study, including clustering algorithms. The results are then examined and the performance is analyzed Finally, the paper presents the limitations of this research work.

III. Literature Review

Over the last few years, the enormous development of big data technology has brought revolution in the tourism industry. Besides, search engines have become primary sources of big data for tourism investigation which track the web searching activities for tourism-related substances. whenever travelers search for travel information through a search engine, they often keep searching traces on the websites. These searching traces are recorded as valuable data and processed to form an important type of big data that reveals people’s attention toward tourism related contents. For instance, ratings/reviews of a tourist destination, hotel availability, distance, accessibility of transports, restaurants nearby etc. Hence this big dataset helps to understand the tourism market [4]. Li et al. presented a comprehensive literature review on different types of big data and application of big data in tourism research [4]. Local governments in many countries are now being more concerned about the tourism sector day by day [5] and they are investing more to highlight their attractions toward the travelers as this industry can be a good source of revenue for any country. According to statista.com, revenue from the Travel Tourism market in Europe is estimated to reach US $107,329m in 2021 whereas 75 % of total revenue will be produced through online sales by 2025 [6].

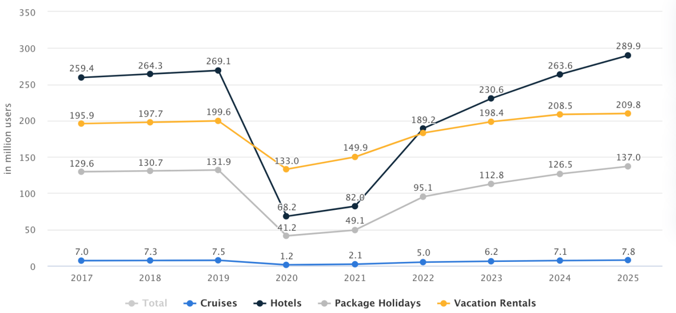


Fig. 2. Statistical data on Travel and Tourism in Europe (Source: statista.com)

In this graph, the y-axis shows the number of travellers in millions at specific regions in Europe for each year. It indicated that every year millions of people from all around the world visit Europe and the number would increase rapidly in coming years as predicted [6]. Chung and Koo discussed both the theoretical and practical influences of VAM (value based adoption model) in the context of social media in tourism. They investigated the travel information search in social media usage using mental accounting theory and found that benefit factors (e.g., information reliability, enjoyment) and sacrifice factors (e.g., complexity, effort) had a remarkable impact on travel information searches. Also, their study reveals that using benefit and sacrifice together can give a broader understanding of an end-user's decision making [7]. In [8], an experiment has been conducted to inspect the effects of different travelling groups' experiences in a hotel in terms of their satisfaction during their stay at that hotel. The authors implemented text mining and content analysis for this study and found that travelers' level of satisfaction highly depends on who they are travel with [8]. [9] proposed a novel method to extract latent dimensions of customer satisfaction from online reviews implementing a data mining approach, latent dirichlet analysis (LDA) where their focus was on quantitative ratings provided on websites. A comparison is shown in [10] where different clustering algorithms have been applied to find the optimal solution that can be implemented as a recommend er system in tourism domain.

IV. Methodology

In our project, after conducting exploratory data analysis (bad format removal, missing value imputation, datatype conversion, and feature engineering) we have implemented three widely used clustering algorithms. We used the modified version of KMeans, the KMeans++, Agglomerative Clustering, and DBScan algorithms. While implementing the algorithms we have also tuned the hyperparameters to improve the performance.

A. *Dataset and Preprocessing*

The dataset has been collected from the UCI machine learning repository. It contains 5456 user ratings from Google reviews. The reviews are done from 25 categories/attractions across Europe. The range of the user rating is from 1 (bad) to 5 (excellent), and the ratings are continuous [10]. We started by conducting the exploratory data analysis on our dataset. Initially, we converted a bad formatted data point to zero which means that place is not visited yet by the user. Then, we converted all the object types to numeric types to fit them into the clustering models. We also conducted missing value imputation by filling the missing values using zero. Additionally, we computed the Pearson correlation heatmap to find out the correlated features. Then, we ran tests by keeping and dropping the correlated features. The correlation heatmap shows that nine features have a moderate correlation (more than 0.5). From which we can make several conclusions. Below are the correlated features, and we kept one from each correlated pair.
i. Beaches Parks
ii. Malls, zoos, restaurants
iii. burger/pizza shops hotels/other lodgings iv. dance clubs swimming pools

B. *KMeans++*

In unsupervised learning, K-Means is one of the most popular algorithms which produces good results with excellent performance. The k-means algorithm clusters the data in n groups with equal variance, and it tries to form the clusters by minimizing the potential function. For this KMeans problem, in the equation (1), k is the number of clusters and it is an integer, X is a set of n data points X R, and C are the centroids. [11]

$$potential\ function = \sum_{x \in X} \min_{c \in C} ||x - c||^2 \qquad (1)$$

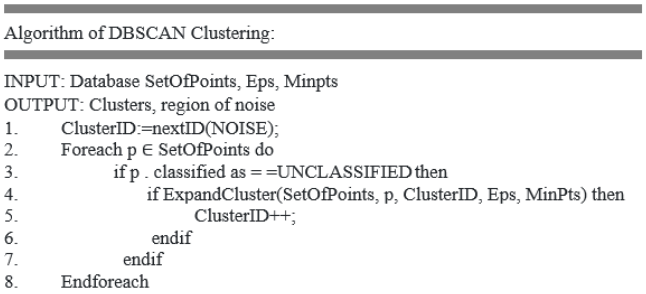
We have selected the augmented version of k-means which is known as k-means ++. The difference between these two algorithms is the selection of the centroids. The algorithm works by choosing the initial center uniformly at random from the dataset. The new center is selected with a probability of equation (2) where the D(x) denotes the shortest distance from a data point to the closest center [11]

$$\frac{D(x)^2}{\sum_{x \in X} D(x)^2} \qquad (2)$$

This weighting process is called the "D2. [2]" This new seeding method performs better and finds better clusters than the previous K-means.

C. *DBScan*

In Unsupervised Learning approach, DB Scan is one of the popular algorithms which can form clusters based on the density of samples available in the space by considering noise data. This algorithm creates clusters by separating areas with high densities from the regions that have low densities. Unlike K means algorithm, the DB scan algorithm forms clusters in arbitrarily shapes that cover most of the noise data. Number of samples available within the given radius defines the density of the cluster. There are two parameters in this model to predict density. The first is the minimum number of samples (min_samples), the minimum number of points required to form a dense region.



```
Algorithm of DBSCAN Clustering:

INPUT: Database SetOfPoints, Eps, Minpts
OUTPUT: Clusters, region of noise
ClusterID:=nextID(NOISE);
Foreach p ∈ SetOfPoints do
        if p . classified as = =UNCLASSIFIED then
            if ExpandCluster(SetOfPoints, p, ClusterID, Eps, MinPts) then
                ClusterID++;
            endif
        endif
Endforeach
```

Fig. 3.

The other parameter is Epsilon value () - distance to define neighborhood eps. Higher no. of min_samples or lower eps value demands greater density to form a cluster. These parameters that helps in tuning by controlling the learning process are called Hyper Parameters. In this approach, every sample falls under any one of the three categories, i.e., core point, border point, or noise point. A core point is the one that has more than or the same number of min_samples within a given radius eps(). Border point is the one that is in the neighborhood of core point and has fewer number of samples than min_samples. A noise point is the one that is not within the radius of eps () and also is not a neighbor for a core point.The following algorithm describes the process of DBSCAN: DBSCAN (algorithm; Martin Ester et al., 1996) [12]

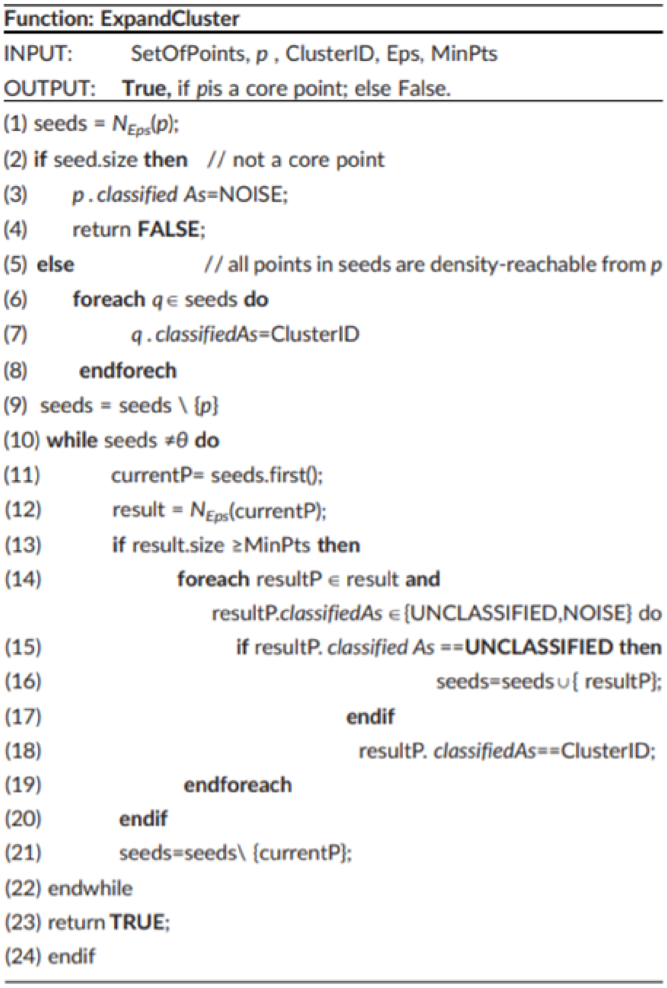
1.Start with an arbitrary point p from the database and retrieve all point densities reachable from p under Eps and MinPts.

2.If p is a core point, the procedure yields a cluster using Eps and MinPts, and the point is classified. Below is the ExpandCluster method that classifes the samples/instances/points into clusters [13]

3. If p is a border point, no points are density-reachable from p, and DBSCAN visits the next unclassified point in the database.

D. Hierarchical Agglomerative Clustering

Hierarchical clustering is one of the main approaches used in data mining to partition a data collection.This method is widely used in terms of social network data analysis. [1]. it produces sequences of data objects by subdividing it. it



```
Function: ExpandCluster
INPUT:     SetOfPoints, p , ClusterID, Eps, MinPts
OUTPUT:    True, if pis a core point; else False.
seeds = $N_{Eps}(p)$;
if seed.size then   // not a core point
    p . classified As=NOISE;
    return FALSE;
else                // all points in seeds are density-reachable from p
    foreach q ∈ seeds do
        q . classifiedAs=ClusterID
    endforeach
    seeds = seeds \ {p}
    while seeds ≠θ do
        currentP= seeds.first();
        result = $N_{Eps}$(currentP);
        if result.size ≥MinPts then
            foreach resultP ∈ result and
                resultP.classifiedAs ∈{UNCLASSIFIED,NOISE} do
                if resultP. classified As ==UNCLASSIFIED then
                    seeds=seeds∪{ resultP};
                endif
                resultP. classifiedAs==ClusterID;
            endforeach
        endif
        seeds=seeds\ {currentP};
    endwhile
    return TRUE;
endif
```

Fig. 4.

continues successively by either splitting larger clusters into smaller or by merging smaller clusters into larger clusters. The outcome of this process is a tree which is called dendrogram.

This method can be classified into two categories: 1. Agglomerative hierarchical clustering 2. Divisive hierarchical clustering Agglomerative: This method follows bottom-up pattern and generates clusters by reducing number of clusters at each level. This process starts with considering each data object as an individual cluster. Gradually the clusters start merging with other clusters based on their similarity and it continues until a termination condition is satisfied or all data object are populated in a single cluster.

Divisive: this methodology begins with a single cluster containing all data objects. Then iteratively each cluster is partitioned into smaller clusters until all data object forms individual cluster. [1][2]

In our study we have implemented the Agglomerative Hierarchical Clustering method for our investigation. This method is attained using an appropriate metric (a measure of distance between pairs of observations), and a linkage criterion. A metric defines the measurement of distance between pairs of clusters. A linkage criterion identifies the dissimilarities of clusters as a function of the pairwise distances of data objects in the clusters.

From the above detailed analysis, we can draw following inferences regarding algorithm performance for this dataset, we can use K-Means when mean of the cluster is defined and we understand that K Means algorithm forms clusters with good quality when huge dataset is used and will not be able to identify outliers. Algorithm is efficient in terms of time taken to form clusters. DBScan algorithm on the contrary can determine what information is categorized as noise/outliers.

DBScan will be able to form clusters of arbitrary shape. We/users are responsible for choosing Parameter values (and Min_samples), slightest change in the parameter settings may lead to different result set. This algorithm has the highest silhouette score which denotes that clusters are formed appropriately. The score is higher when clusters are dense and well separated, which relates to a standard concept of a cluster.

Agglomerative Clustering is taking more time when compared K- Means and DBScan and has very low silhouette. We conclude that Agglomerative Clustering has very low performance when compared with the other algorithms on chosen metrics (Execution time, Silhouette score, Calinski Harabasz Score). The efficiency of clustering algorithms can be improved by removing the limitations of the clustering techniques.

V. Experimental Results

In the experimental setup, the results obtained after running the clustering techniques for Travel Review data set which consists of 5456 instances and 24 attributes, such as churches, resorts, beaches, parks, theatres, parks, Zoo, museums, pubs, malls, etc. To construct the algorithms, we use Python3 IDE in Google Compute Engine (Google Collab), Scikit Learn Libraries as Knowledge sources. This experiment is performed on Duo Core with 2.20 GHz CPU and 12GB RAM running on Linux Platform. The result for each clustering algorithms is shown and described below.

A. KMeans++:

To find out the optimal number of clusters we have implemented the elbow method by computing the sum of the squared distance between centroid and each member of the cluster for twenty iterations. Based on the elbow plot we picked two variations for KMeans++, five and ten.

Table 1. shows the result of K-Means clustering algorithm on the Travel Review dataset for different number of clusters. Figure 1 shows that when the number of clusters increased, the inertia has decreased. From Fig 1 and Table 1. we concluded that Kmeans++ clustering algorithm performs well in terms of execution time and Silhouette score when K value is 10 after feature engineering i.e with 19 Attributes. Table 2 shows the result of DBSCAN clustering algorithm on the Travel Review dataset which contains 5456 instances and 24 attributes for different () and “Min_Samples” parameters.

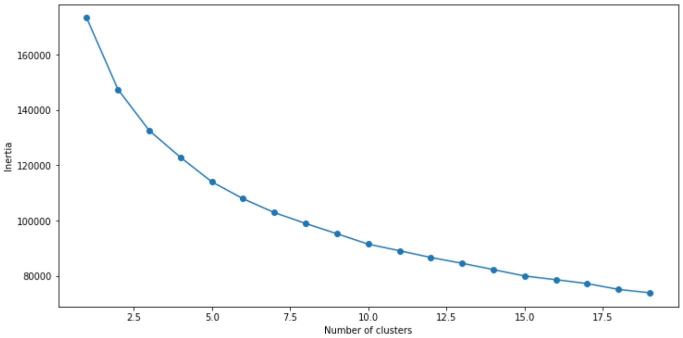
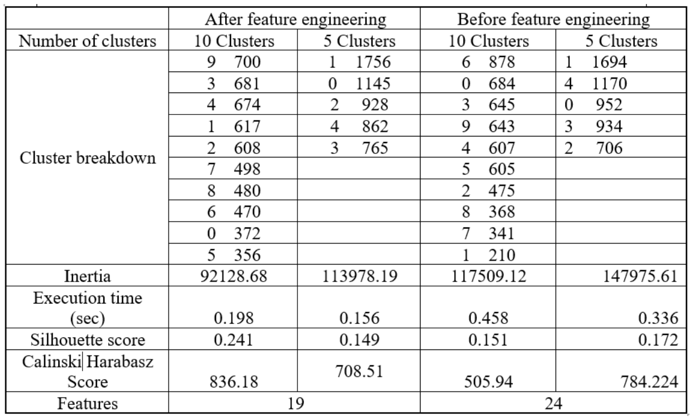


Fig. 5.



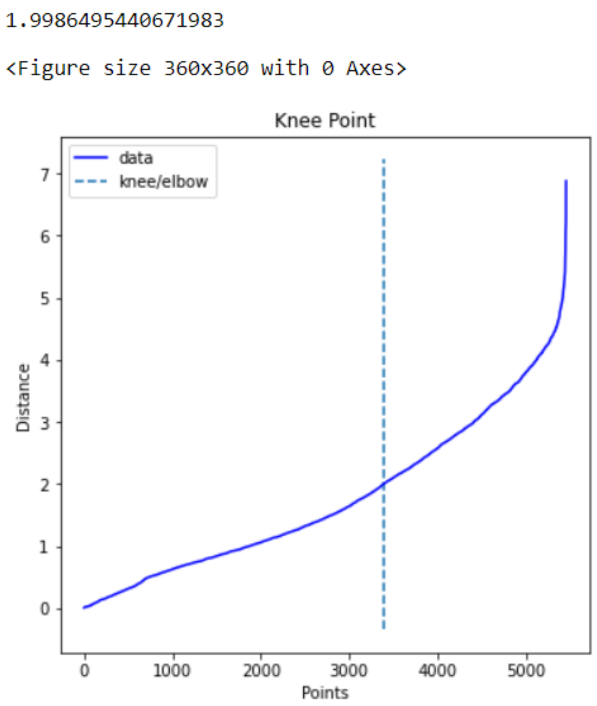
	After feature engineering		Before feature engineering	
Number of clusters	10 Clusters	5 Clusters	10 Clusters	5 Clusters
Cluster breakdown	9 700	1 1756	6 878	1 1694
	3 681	0 1145	0 684	4 1170
	4 674	2 928	3 645	0 952
	1 617	4 862	9 643	3 934
	2 608	3 765	4 607	2 706
	7 498		5 605	
	8 480		2 475	
	6 470		8 368	
	0 372		7 341	
	5 356		1 210	
Inertia	92128.68	113978.19	117509.12	147975.61
Execution time (sec)	0.198	0.156	0.458	0.336
Silhouette score	0.241	0.149	0.151	0.172
Calinski Harabasz Score	836.18	708.51	505.94	784.224
Features	19		24	

Fig. 6.

B. DBScan:

In order to tune these hyperparameters of DBScan algorithm, first, we need to find the optimal eps value. We tried two methods to find the optimal eps value. We found the nearest neighbor distances in the first method and plotted the minimum distances that form an elbow curve with the optimal eps value() and sample points density. We have used the “Nearest Neighbhours” function to find the minimum distances and Knee Locator to identify the inflection value as shown in the figure 2.

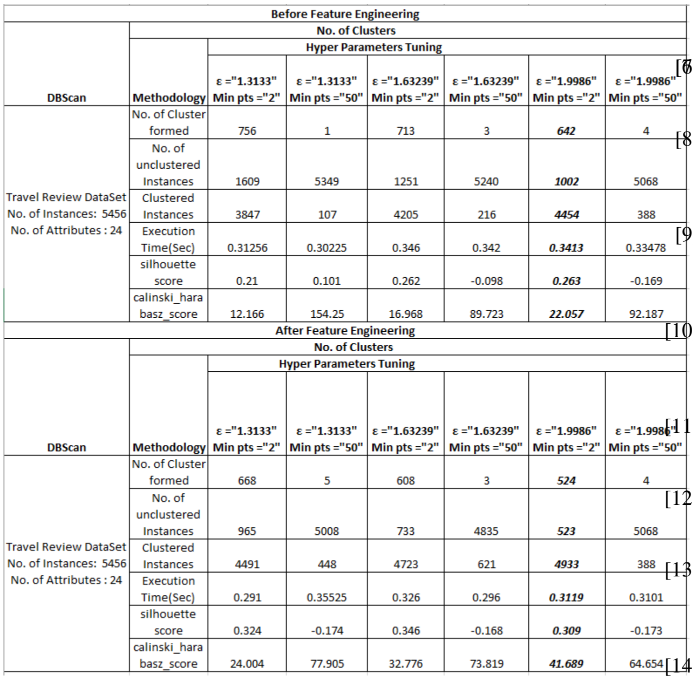
Table 2. shows the result of DB Scan clustering algorithm on the Travel Review dataset for different number of clusters. Figure 3 shows how the silhouette score is affected when there is a change in Min_samples considered over a optimal eps value (). As per the Figure 2 above and Table 2, the optimal value for eps is 1. 9986. Using this eps value identified, we have estimated that min_samples = 2 has the high silhouette score. Alternatively, we have manually looped through a definite range of eps (ranging from 0.5 to 2.0 with a step size of 0.01) and min_samples (ranging from 1 to 50 with a step size of 1) and estimated the combination that has highest silhouette score and this result matched with the first method (Elbow method). From Figure 2, Table 2 and second method



```
1.9986495440671983

<Figure size 360x360 with 0 Axes>
```

Fig. 7.



Before Feature Engineering							
	No. of Clusters						
		Hyper Parameters Tuning					
DBScan	Methodology	ε ="1.3133" Min pts ="2"	ε ="1.3133" Min pts ="50"	ε ="1.63239" Min pts ="2"	ε ="1.63239" Min pts ="50"	ε ="1.9986" Min pts ="2"	ε ="1.9986" Min pts ="50"
Travel Review DataSet No. of Instances: 5456 No. of Attributes : 24	No. of Cluster formed	756	1	713	3	***642***	4
	No. of unclustered Instances	1609	5349	1251	5240	***1002***	5068
	Clustered Instances	3847	107	4205	216	***4454***	388
	Execution Time(Sec)	0.31256	0.30225	0.346	0.342	***0.3413***	0.33478
	silhouette score	0.21	0.101	0.262	-0.098	***0.263***	-0.169
	calinski_harabasz_score	12.166	154.25	16.968	89.723	***22.057***	92.187
After Feature Engineering							
	No. of Clusters						
		Hyper Parameters Tuning					
DBScan	Methodology	ε ="1.3133" Min pts ="2"	ε ="1.3133" Min pts ="50"	ε ="1.63239" Min pts ="2"	ε ="1.63239" Min pts ="50"	ε ="1.9986" Min pts ="2"	ε ="1.9986" Min pts ="50"
Travel Review DataSet No. of Instances: 5456 No. of Attributes : 24	No. of Cluster formed	668	5	608	3	***524***	4
	No. of unclustered Instances	965	5008	733	4835	***523***	5068
	Clustered Instances	4491	448	4723	621	***4933***	388
	Execution Time(Sec)	0.291	0.35525	0.326	0.296	***0.3119***	0.3101
	silhouette score	0.324	-0.174	0.346	-0.168	***0.309***	-0.173
	calinski_harabasz_score	24.004	77.905	32.776	73.819	***41.689***	64.654

Fig. 8.

where we manually looped through. We conclude that DB Scan clustering algorithm results in high Silhouette score and less execution time, when the eps () value is “1.9986” and Min_samples/Min_pts value is “2”. Table III shows the result of Agglomerative clustering algorithm on the Travel Review dataset for different number.

C. *Agglomerative Clustering:*

There are five categories of metrics that can be used to compute the linkage. They are: “euclidean”, “l1”, “l2”, “manhattan”, “cosine”, or “precomputed”. And, 4 types of linkages are there to decide which distance to apply between sets of observation. Linkage=’ward’, ‘complete’, ‘average’, ‘single’ We have applied linkage=’ward’ and metric=’euclidean’ for our analysis [3].

REFERENCES

[1] Google trend. Available at https://trends.google.com/trends/explore?date=2016-01-01 2019-12-31q=tourist destination in europe. [Online; accessed 17-October-2021].

[2] Yue Guo, Stuart J Barnes, and Qiong Jia. Mining meaning from online ratings and reviews: Tourist satisfaction analysis using latent dirichlet allocation. Tourism management, 59:467–483, 2017.

[3] Roger Hallowell. The relationships of customer satisfaction, customer loyalty, and profitability: an empirical study. International journal of service industry management, 1996.

[4] J. Li, L. Xu, L. Tang, S. Wang and L. Li, ”Big data in tourism research: A literature review”, Tourism Management, vol. 68, pp. 301-323, 2018. Available: 10.1016/j.tourman.2018.03.009.

[5] J. Xu and J. Xu, ”Research on the Design of Online Travel Service Recommendation System Based on Data Analysis”, Lecture Notes on Data Engineering and Communications Technologies, pp. 1042-1046, 2021. Available: 10.1007/978-981-16-5857-$0_132[Accessed 20 October 2021].”Travel Tourism - Europe|Statista Market Forecast”, Statista, 2021.[Online].Available : https : //www.statista.com/outlook/mmo/travel - tourism/europeusers.[Accessed : 20 - Oct - 2021]$.

[6] N. Chung and C. Koo, ”The use of social media in travel information search”, Telematics and Informatics, vol. 32, no. 2, pp. 215-229, 2015. Available: 10.1016/j.tele.2014.08.005.

[8] D. Ahn, H. Park and B. Yoo, ”Which group do you want to travel with? A study of rating differences among groups in online travel reviews”, Electronic Commerce Research and Applications, vol. 25, pp. 105-114, 2017. Available: 10.1016/j.elerap.2017.09.001.

[9] Y. Guo, S. Barnes and Q. Jia, ”Mining meaning from online ratings and reviews: Tourist satisfaction analysis using latent dirichlet allocation”, Tourism Management, vol. 59, pp. 467-483, 2017. Available: 10.1016/j.tourman.2016.09.009.

[10] S. Renjith, A. Sreekumar and M. Jathavedan, ”Evaluation of Partitioning Clustering Algorithms for Processing Social Media Data in Tourism Domain,” 2018 IEEE Recent Advances in Intelligent Computational Systems (RAICS), 2018, pp. 127-131, doi: 10.1109/RAICS.2018.8635080.

[11] D. Arthur, S. Vassilvitskii, “K-means++: The advantages of careful seeding,” Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms..

[12] Jyo1 Xu, X., Ester, M., Kriegel, H. and Sander, J., 1996. Clustering and knowledge discovery in spatial databases. Vistas in Astronomy, 41(3), pp.397-403.

[13] O. Limwattanapibool and S. Arch-int, ”Determination of the appropriate parameters for K-means clustering using selection of region clusters based on density DBSCAN (SRCD-DBSCAN)”, Expert Systems, vol. 34, no. 3, p. e12204, 2017. Available: 10.1111/exsy.12204 [Accessed 21 October 2021].

[14] H. Hexmoor, ”Diffusion and Contagion”, Computational Network Science, pp. 45-64, 2015. Available: 10.1016/b978-0-12-800891-1.00006-8 [Accessed 20 October 2021].

[15] M. Halkidi, ”Hierarchial Clustering”, Encyclopedia of Database Systems, pp. 1291-1294, 2009. Available: 10.1007/978-0-387-39940-9_604 [Accessed 21 October 2021].

[16] ”sklearn.cluster.AgglomerativeClustering”,scikit-learn, 2021. [Online]. Available:https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.htmlsklearn.cluster.AgglomerativeClustering. [Accessed: 21- Oct- 2021].