Determining the Optimal Number of Clusters using Silhouette Score as a Data Mining Technique

https://doi.org/10.3991/ijoe.v19i04.37059

Ylber Januzaj[1], Edmond Beqiri[1](✉), Artan Luma[2]
[1]Faculty of Business, University "Haxhi Zeka", Peja, Kosovo
[2]CST Faculty, South East European University, Tetovo, Northern Macedonia
edmond.beqiri@unhz.eu

Abstract—The identification of the same objects is very important in determining the similarity between different objects. Nowadays, there are several techniques that allow us to divide objects into different groups that differ from one to another. In order to have the best separation between the clusters, it is required that the optimal determination of the number of clusters of a corpus be made in advance. In our research, the Silhouette score technique was used in order to make the optimal determination of this number of clusters. The application of such a technique was done through the Python language, and a corpus of unstructured job vacancy data was used. After determining the optimal number, at the end we present these clusters and the similarity between them, this presentation will be done in the form of a graph in a suitable format.

Keywords—Silhouette score, clusters, data mining, corpus, job vacancy

1 Introduction

The dynamic growth of data from second to second has made their processing even more challenging in terms of extracting different analyses [1]. Different fields have progressed based on the analyzes that Data Mining enables us. One of the biggest challenges of Machine Learning is the processing of data that does not have a class label which is known as unsupervised learning [2]. Through unsupervised learning, distributed modeling is enabled so that we have more information on the data being processed. One of the most preferred forms is clustering, which allows us to detect different groups within a corpus of data [3]. These groups contain objects that are very similar to each other within the same group, while they have a great distinction with the objects of other groups.

Clustering has found application in many different fields, and through it the analysis of data that has been impossible to analyze through other techniques has been made possible [4]. Some of the fields where Clustering has found application are: bioinformatics, medicine, social sciences, computer sciences, etc.

In our research, we analyze our corpus in order to determine the optimal or most appropriate number of clusters that our corpus will contain. Later technique that will be used is k-means clustering method, where is required to determine the number of clusters to be created, and an analysis in order to determine the optimal number of clusters to be created is necessary. There are a number of methods used to analyze clusters and to determine the optimal number of clusters, and of course, none yields the same results as it depends on the method used. The methods used to analyze the optimal number of clusters are: *Silhouette method, Elbow method, R analysis, Gap statistic method.*

All of the methods mentioned above are methods used to analyze the optimal number of clusters. Of course, there is no single method used to analyze the cluster, but it depends on the form of the data and the method in terms of its accuracy and appropriateness.

All of the above-mentioned methods are used to determine the optimal number of clusters and some of them are also used for statistical research.

In our case, we use the silhouette method as the most widespread method in order to determine the optimal number of clusters that we divide into our corpus. Later these clusters will be compared to each other.

2 Silhouette analysis

The method which shows how close an object is to its cluster compared to the other cluster is known as Silhouette analysis. According to [5], the average obtained by silhouette analysis shows exactly how optimal the number of clusters created is. A higher average indicates that the number k of the clusters is optimal and at such numbers, it is preferable to divide the corpus with textual content [6,7].

The values that can be obtained after applying the Silhouette analysis range from –1 to +1. The higher the value, the closer the object is to its cluster, and vice versa, the smaller the value that is acquired, the farther away is the object with its cluster. After calculating these values, an average is obtained which shows the optimal number of clusters. This number is very easy to assign, since the number where we get the highest average is the optimal number that our cluster will contain.

Whether or not an object is aligned with its cluster can be measured in several forms, but in the case of silhouette analysis, this is done using the Euclidean distance method, by calculating the two points that are placed in Euclidean space.

Since in our case we are dealing with textual data, we use the k-means method as a cluster method. Below we present the mathematical calculations that are used to perform the Silhouette analysis to proceed later with its application through algorithms and the Python language in order to perform the optimal number calculations of our corpus.

According to [4], we present a case where we have created some clusters which we present with C_M and C_N, and compare the distance *a* between O_i and other objects in C_M, and the distance *b* between O_i and other objects in C_N.

$$a(O_i) = \frac{1}{|C_M| - 1} \sum_{O_j \in C_M,\, O_{j \neq} O_i} d(O_i, O_j) \tag{1}$$

$$b(O_i) = minC_N \neq C_M \frac{1}{|C_B|} \sum_{O_j \in C_N} d(O_i, O_j) \tag{2}$$

$$silhouette(O_i) = \frac{b(O_i) - a(O_i)}{max\{a(O_i), b(O_i)\}} \tag{3}$$

Above we have presented the mathematical equations, which calculate the average silhouette. As we can see in (Eq. (1,2,3)), three equations have been presented which contain the calculation steps, starting from the first step presented in the first equation. According to this equation, $a(O_i)$ is equal to the division between 1 and the absolute value of the cluster C_M minus 1 and the sum of the distance between O_i and O_j where both are objects of a cluster, but must not be equal to each other.

Once the first cluster is computed, we must define the second cluster which in our case is $b(O_i)$. This cluster is the minimum distance of one of the objects of the first cluster, but it must never be part of the first cluster. As we can see in the second equation, it is equal to the minimum of the first cluster that is different from the second cluster. Then this minimum value is multiplied by 1 partition for the absolute value of C_N, which in this case is the second cluster. And this value is also reduced by the sum of the distance between the objects O_i and O_j, where O_j is an element of the second C_N cluster.

Once the second cluster is defined, in the third equation we do the silhouette calculation.

According to the third equation, the silhouette equals the division between the subtraction of $b(O_i)$ and $a(O_i)$ and the maximum value between the first cluster $a(O_i)$ and the second cluster $b(O_i)$.

If we want to calculate the classification quality of a single object, then we can extend the last silhouette equation. Below we present the case of calculating the quality of all objects that are part of a cluster, as well as the case of calculating the quality of clusters one by one.

$$Silhouette(C_i) = \frac{1}{|C_i|} \sum_{O_j \in C_i} silhouette(O_j) \tag{4}$$

In (Eq. (4)) is presented the case of calculating the quality of all objects that are part of a cluster. As can be seen in the equation, the silhouette equals the division of the value 1 and C_i, and the multiplication of this value by the sum of the silhouette objects (O_j), where the object (O_j) must be an element of the first cluster, while calculating the quality value for each cluster according to the equation below.

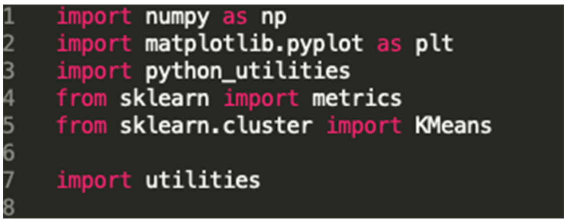
$$Silhouette(C) = \overline{silhouette(m)} = \frac{1}{m} \sum_{i=1}^{m} silhouette(C_i) \tag{5}$$

In (Eq. (5)) is presented the calculation of the quality value for each cluster individually. As we can see, *silhouette*(C) is equal to *silhouette*(m) which is a vinculum, that is, the set of all cluster values. And this value is equal to the division between 1 and m cluster, and the output of this value is the sum of the *silhouette*(C_i), where *i* starts from 1 to *m* which is the number of clusters that are defined in the system.

So, as can be seen, the calculation of the quality of the cluster and the objects that are part of the cluster, through silhouette analysis, can be done in a very precise way through the mathematical calculations as mentioned earlier presented by Kaufman and Rousseeuw.

3 Explanation of silhouette score through python example

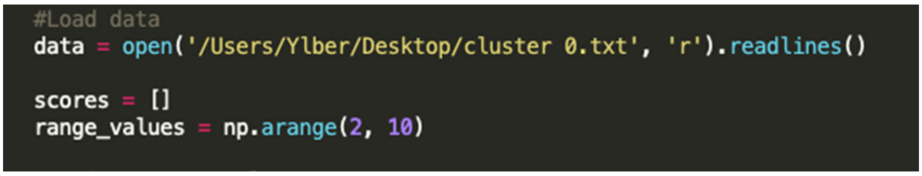
After the mathematical equations that make up the silhouette are presented, we present this analysis through the algorithm, which will be constructed in the Python language. Once we have created the algorithm, we apply it to our corpus, to see what the optimal number of clusters we need to create in our system is, to proceed later with comparing syllabuses with each cluster, and comparing all clusters with each other. The comparison of these clusters will be made using our model in order to determine the similarity of textual content between these clusters.



```
import numpy as np
import matplotlib.pyplot as plt
import python_utilities
from sklearn import metrics
from sklearn.cluster import KMeans

import utilities

```

Fig. 1. Importation of libraries for silhouette analysis

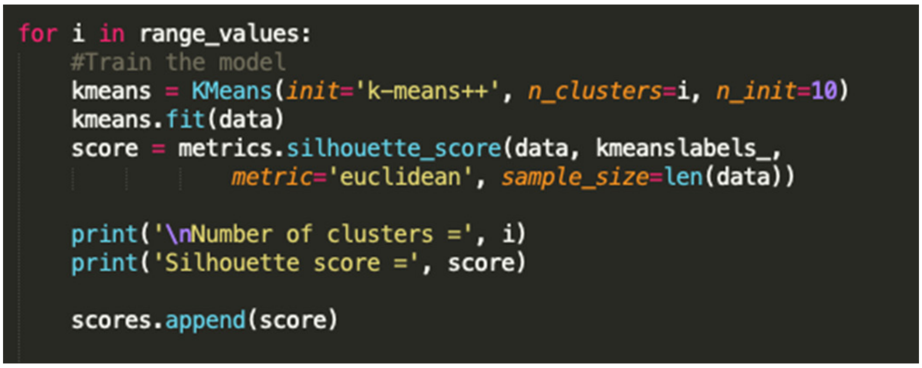
Figure 1 shows the part of the bookstore import we need in order to implement silhouette analysis. As can be seen, there are all the libraries that are needed starting from the mathematical calculations that the system will do to those needed to create the figures with the results obtained.



```
#Load data
data = open('/Users/Ylber/Desktop/cluster 0.txt', 'r').readlines()

scores = []
range_values = np.arange(2, 10)
```

Fig. 2. Load data of vacancy corpus

In Figure 2 is shown the load data of vacancy corpus, which is the data that has been processed and prepared for this part. The data to be imported is the data that was originally converted to vector values. After this part, it will be the model training part.

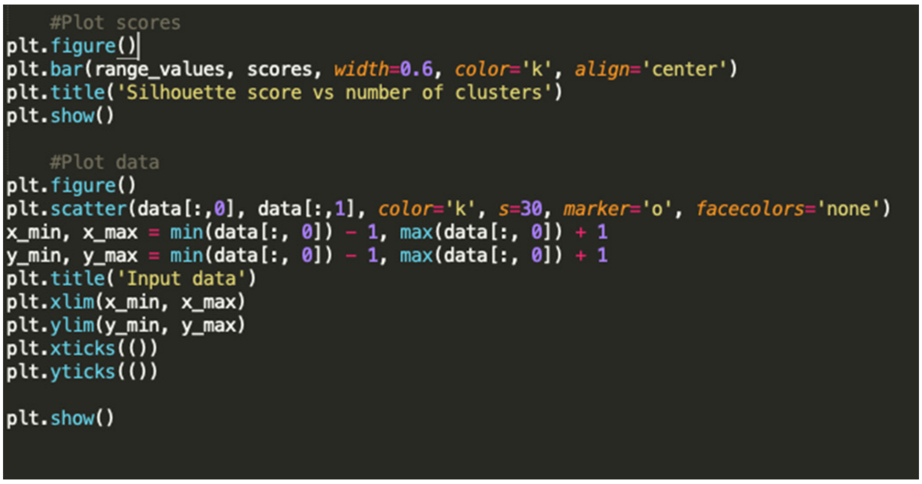


```
for i in range_values:
    #Train the model
    kmeans = KMeans(init='k-means++', n_clusters=i, n_init=10)
    kmeans.fit(data)
    score = metrics.silhouette_score(data, kmeanslabels_,
                metric='euclidean', sample_size=len(data))

    print('\nNumber of clusters =', i)
    print('Silhouette score =', score)

    scores.append(score)
```

Fig. 3. Train model of silhouette analysis

Figure 3 shows the training part of our model, which will do the silhouette analysis. As can be seen in Figure 3, this section defines the type of clusters that in this case is k-means, as well as the definition of the score in our case is silhouette, and the measurement of the distance between objects that are within clusters will be done with Euclidean distance.

After defining the clusters and the score, the results are finally printed in textual form, such as the number of clusters, as well as the silhouette score, defined above.



```
    #Plot scores
plt.figure()
plt.bar(range_values, scores, width=0.6, color='k', align='center')
plt.title('Silhouette score vs number of clusters')
plt.show()

    #Plot data
plt.figure()
plt.scatter(data[:,0], data[:,1], color='k', s=30, marker='o', facecolors='none')
x_min, x_max = min(data[:, 0]) - 1, max(data[:, 0]) + 1
y_min, y_max = min(data[:, 0]) - 1, max(data[:, 0]) + 1
plt.title('Input data')
plt.xlim(x_min, x_max)
plt.ylim(y_min, y_max)
plt.xticks(())
plt.yticks(())

plt.show()
```

Fig. 4. Plot data and plot scores definition

In Figure 4 we can see the silhouette score calculation algorithm, the creation of figures that make the visual representation of the silhouette score we obtained. These figures have been created through the libraries we created at the beginning of our algorithm, and as can be seen in the first section, bar charts have been created that represent the relationship between the silhouette score and the number of clusters. This section also shows the optimal number of clusters in graphical form, where we can see what the result of all the clusters created by our system is. In our case, this model will be able to graphically represent the optimal number of clusters created by the system, to continue with further analysis of these clusters.

4 Main results

In addition to showing the bar charts between the silhouette score and the number of clusters, this algorithm will also display the objects of the clusters that are part of each cluster. Here one can see which cluster is closest to the other, as well as the distance of each object, which in this case is calculated by Euclidean distance. After defining the algorithm and preparing the data, we are ready to execute the algorithm which we will present using the figures below.

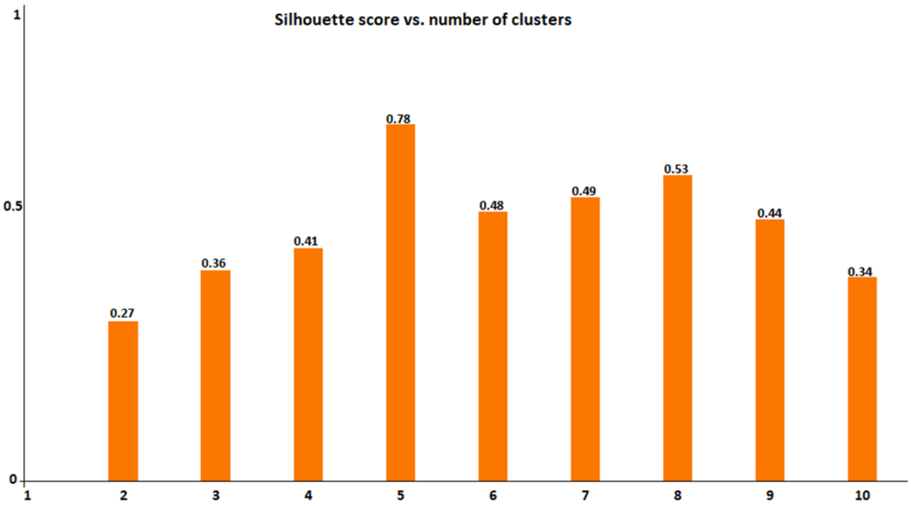


Fig. 5. Silhouette score versus number of clusters

Figure 5 shows the number of clusters relative to the silhouette score that each cluster has. As we can see in this graph, the number of clusters is from **2** to **10**. The smallest silhouette score reached by the cluster set is **0.27** where we have two clusters. Then the value of silhouette score for 3 clusters is **0.36**, for 4 clusters we have the value of **0.41**. The highest value is with 5 clusters, where the silhouette score reaches **0.78**, which represents the optimal number of clusters that our research should contain. For 6 clusters we have a decrease in silhouette score, where its value reaches **0.48**, and for 7 clusters we have **0.49**. As for the last two groups with 9 and 10 clusters, we have values of **0.44** and **0.34** respectively.

Such an analysis helps us a great deal to determine the number of clusters that our research will contain. According to this analysis, our labor market demand corpus will contain 5 clusters, as it is the highest value of the silhouette score which is achieved by our model.

Since we have the optimal number of clusters, we will then use these 5 clusters to compare them with the university syllabuses that are part of the analysis, but also to compare the textual similarity between all the clusters created. Below we present the graph of the full score silhouette, based on which we will be able to see which cluster is closer to each other, and which cluster objects are more aligned with other cluster objects.

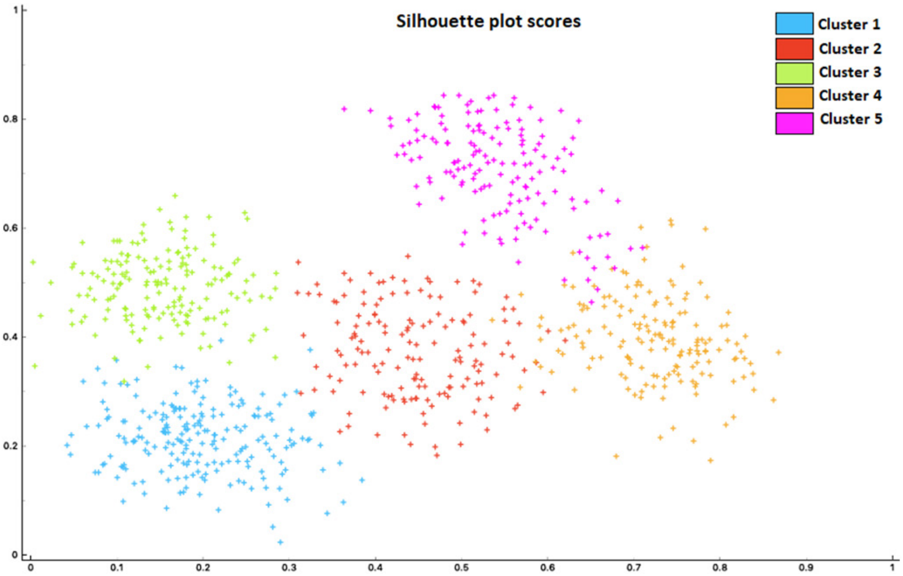


Fig. 6. Silhouette plot score

Figure 6 shows the full score silhouette after running the algorithm we built for such an analysis. As can be seen in Figure 6, only the optimal number of clusters defined by the system has been shown here. So, we have 5 clusters that are presented in the x and y dimensions, where we can even guess which, one is closest to the other. According to the graph, we have the color split of all clusters, from 1 to 5, and we can also see which cluster is closest to the other. According to the graph, we have proximity between the objects of **cluster 1** and **cluster 2**, since the distance between them is very small, but they are not equal to each other. There is also a small distance between **cluster 1** and **cluster 3**, and between **cluster 2** and **cluster 3**. We also have a small distance between **cluster 2** and **cluster 5**, as well as **cluster 3** and **cluster 5**, while there are large distances between **cluster 5** and **cluster 1** objects, as well as **cluster 5** and **cluster 2**. According to the graph that made the representation of objects between clusters, there is no proximity between **cluster 5** objects and **cluster 1**, and **cluster 5** and **cluster 3**.

5 Conclusion

In our research, we have presented the silhouette score analysis as one of the most appropriate analyzes to determine the optimal number of clusters that a corpus should contain.

First, we presented the technical form of how this analysis works, and through the equations, the usual form of silhouette score analysis was presented. It was necessary to present the form of mathematical calculation, to continue later with the implementation of the computer algorithm to analyze the corpus.

The implementation of the computer algorithm was done through the Python language, where the ready-made silhouette score libraries were first imported, to continue later with the training of the model and the graphic presentation of the results after the silhouette score.

At the end, we presented the results of the analysis which were made after the execution of the computer algorithm.

These results show the optimal number of clusters that our corpus should contain. After presenting the optimal number, a comparison was made between the clusters, where we saw the greatest similarity and the least similarity between the clusters.

Finally, based on algorithm calculations, we have presented graphically a sketch which shows 5 numbers of clusters, each of them with different groups. These groups have different objects which are similar to each other inside the group, but different with objects outside the group.

And based on this we can conclude that the silhouette score is a very powerful technique for determining the optimal number of clusters.

6 References

[1] Aziz, A., Yusof, Y., (2016). "Graduates Employment Classification using Data Mining Approach". In *Proceedings of the International Conference on Applied Science and Technology*, ICAST. https://doi.org/10.1063/1.4960842

[2] Sahu, S., Bhatt, M., (2017). "Big Data Classification of Student Result Prediction". *In International Journal of Research In Science & Engineering*, *3*(2).

[3] Ahmad, F., Ismail, N. H., Aziz, A. A., (2015). "Using Classification Data Mining Techniques". In *Applied Mathematical Sciences*, *9*(129), pp. 6415–6426. https://doi.org/10.12988/ams.2015.53289

[4] Mayra, Z. R., Cezar, M. C., (2019). "Clustering Algorithms: A Comparative Approach". *PLoS ONE, 14*(1): e0210236. https://doi.org/10.1371/journal.pone.0210236

[5] Salihoun, M. (2020). State of Art of Data Mining and Learning Analytics Tools in Higher Education. *International Journal of Emerging Technologies in Learning (iJET)*, *15*(21), pp. 58–76. https://doi.org/10.3991/ijet.v15i21.16435

[6] Chen, Z., Liu, L., Qi, X., Geng, J. (2016). Digital Mining Technology-Based Teaching Mode for Mining Engineering. *International Journal of Emerging Technologies in Learning (iJET)*, *11*(10), pp. 47–52. https://doi.org/10.3991/ijet.v11i10.6271

[7] Boulaajoul, M., Aknin, N. (2019). The Role of the Clusters Analysis Techniques to Determine the Quality of the Content Wiki. *International Journal of Emerging Technologies in Learning (iJET)*, *14*(01), pp. 150–158. https://doi.org/10.3991/ijet.v14i01.9074

7 Authors

Dr. Ylber Januzaj is an Assistant Professor in University "Haxhi Zeka", Peja, Faculty of Business, Kosovo. He is Professor in the area of Informatics. He holds a PhD diploma on E-Technologies. His research interests are: Machine Learning, Computer Networks, and Database. (ylber.januzaj@unhz.eu)

Dr. Edmond Beqiri is a Full Professor in University "Haxhi Zeka", Peja, Faculty of Business, Kosovo. He is Professor in the area of Information of Technology. He holds a PhD diploma on Computer Sciences. His research interests are: Information Technology, E-Business, and Information Systems. (edmond.beqiri@unhz.eu)

Dr. Artan Luma is a Full Professor in South East European University, CST Faculty, Tetovo, Northern Macedonia. He holds a PhD diploma on Computer Sciences. His research interests are: Computer security, Networking. (a.luma@seeu.edu.mk)

Article submitted 2022-11-27. Resubmitted 2023-01-29. Final acceptance 2023-01-29. Final version published as submitted by the authors.