K-means Algorithm on Heart Failure Survival from Serum Creatinine and Ejection Fraction Dataset

Dibbyo Saha, Computer Science, Toronto Metropolitan University

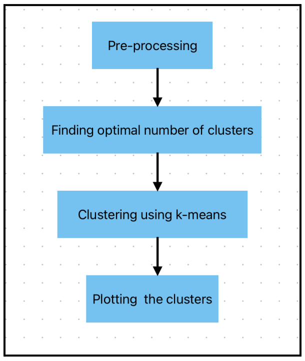
Date: December 6th, 2023

Background

The original dataset comprises the heart failure clinical records from 299 patients collected in 2015 emphasized by 12 attributes and 1 target column respectively, namely: ‘age’, ‘anaemia’, ‘creatinine_phosphokinase’, ‘diabetes’, ‘ejection_fraction’, ‘high_blood_pressure’, ‘platelets’, ‘sex’, ‘serum_creatinine’, ‘serum_sodium’, ‘smoking’, ‘time’, and (target) ‘death_event’. The dataset was collected from UCI Irvine Machine Learning Repository and the data mostly represents a set of multivariate data of either integer or float data type. This rich set of bioinformatics data entails essential and interesting insights about the impact on the survival of heart failure patients from serum creatinine and ejection fraction, as well as crucial information that can be used to rank attributes that impact the survival of heart failure patients.[1][2]

Methods

All the technical analysis and implementation of the algorithm was performed in *Python* along with its various machine learning modules like *sklearn* and data processing libraries. Firstly the entire dataset in the *heart_failure_clinical_records_dataset.csv* file was loaded into a data frame using *pandas* and preprocessed to replace any missing value with the median value of the respective column. Methods like *head()* and *dtypes* were used to get more insight into the dataset’s structure to plan the next few steps to construct the algorithm. It was preferred not to replace it with the mean value to avoid relying on the assumption that the data represents a normal distribution. The data frame was then scaled and normalized employing the *StandardScaler()* and *fit_transform()* methods in the *preprocessing* library in the *sklearn* module (a machine learning module in *Python*). A new data frame was created to extract the *serum_creatinine* and *ejection_fraction* using the *values* method for the original data frame. Next, the *silhouette_score* method from the *metrics* library was used to determine the *k-value* with the given number of clusters to determine the optimal number of clusters. Furthermore, *KMeans* was used and *fit_predict* was used on the data frame to create k-means clustering as well as *cluster_centers_* to create centroids for the clusters. In addition to that, *PCA* was used to structure the data into two dimensions for ease in visualization purposes followed by *fit_transform* and *transform* to scale and transform the dataset and centroids respectively. Finally, *pyplot* was used to plot and visualize the resulting clusters and corresponding centroids as plots (Figure 2). [3][4][5][6]



Pre-processing

Finding optimal number of clusters

Clustering using k-means

Plotting the clusters

Figure 1: Simplified structure of the algorithm as a flowchart.

Results

After running the *silhouette_score* method, the corresponding Silhouette scores were noted and *k=3* was picked for its highest score.

k	*Silhouette Score*
2	0.46090358321807523
3 (Optimal k)	**0.5024395254100789**
4	0.44656679677254907
5	0.47980073144546537
6	0.45967156981022
7	0.4787069816264261
8	0.47972617342611823
9	0.48012323326380535
10	0.4669169898416124

Table 1: Results of respective k-values and Silhouette Score.

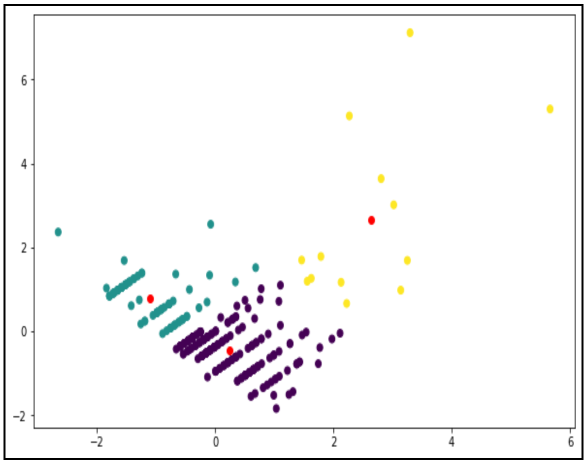


Figure 2: Cluster plotting from the results of the k-means algorithm for optimal k, red points are centroids, and the other colors represent distinguished cluster points.

Conclusions

From Figure 2, it can be observed that data points have been grouped as distinguished cluster points around their respective centroids (red points). Due to the optimal k value being 3, three distinct clusters have been plotted based on attributes ‘serum_creatinine’ and ‘ejection_fraction’. Some outliers can also be observed especially in the case of the yellow cluster points which are further separated from their respective centroid. However, the clusters are fairly well-separated from each other with most of the points strongly near the centroids.

References

[1] UCI Machine Learning Repository, "Heart Failure Clinical Records", [Online]. Available: https://archive.ics.uci.edu/dataset/519/heart+failure+clinical+records. [Accessed: 6th December, 2023].

[2] D. Chicco and G. Jurman, "Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone," Semantic Scholar, [Online]. Available: https://www.semanticscholar.org/paper/Machine-learning-can-predict-survival-of-patients-Chicco-Jurman/e64579d8593140396b518682bb3a47ba246684eb. [Accessed: 6th December, 2023].

[3] W3Schools, "Python Machine Learning - K-means," [Online]. Available: https://www.w3schools.com/python/python_ml_k-means.asp. [Accessed: 6th December, 2023].

[4] GeeksforGeeks, "Determining the Number of Clusters in Data Mining," [Online]. Available: https://www.geeksforgeeks.org/determining-the-number-of-clusters-in-data-mining/. [Accessed: 6th December, 2023].

[5] A. Geron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd ed. O'Reilly Media, pp. 237-255.

[6] A. C. Muller and S. Guido, Introduction to Machine Learning with Python. O'Reilly Media.