Jurma : Jurnal Program Mahasiswa Kreatif
Universitas Ibn Khaldun Bogor, Indonesia
Vol. 9, No. 1, Juni 2025, pp. 50~55
ISSN: 2615-8019, DOI: 10.32832/pkm
http://pkm.uika-bogor.ac.id/index.php/pkm-p

Flower Species Grouping to Find Out Outliers Using DBSCAN Clustering on Google Colab

Adzkia Nur Nasution *, Ardilla Syafitri Lubis, Keysa Shifa Adwitia Sitepu, Rezkya Nadilla Putri, Arnita
Fakultas Matematika Dan Ilmu Pengetauan Alam, Universitas Negeri Medan
Jl. William Iskandar Ps. V No.104, Kenangan Baru, Kec. Percut Sei Tuan, Kabupaten Deli Serdang, Sumatera Utara 20371, Indonesia

Article Info

Article history:

Received November 18, 2024
Revised December 23, 2024
Accepted January 16, 2025

Keywords:

DBSCAN Clustering
Flower Species Clustering
Google Colab
Outlier Detection

ABSTRACT

This study aims to identify iris flower species and detect outlier data using the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) grouping method on the Google Colab platform. The data used were iris datasets from the UCI Machine Learning Repository, which consisted of three species: Setosa, Versicolor, and Virginica, with attributes such as sepal length and width and petals. In this study, the DBSCAN process includes the pre-processing stage of data, parameter determination, model building, and visualization of clustering results. DBSCAN was chosen because it is able to detect outliers and does not require a predetermined number of clusters, making it effective for irregular data. The results showed that DBSCAN managed to group the data into three main clusters, with clear identification of outliers. Cluster 0 includes all Setosa data, while cluster 1 consists of Versicolor and Virginica data. The -1 cluster, which contains data that is considered an outlier, suggests that some specimens have unusual characteristics. In conclusion, the DBSCAN method is effective in grouping iris flower data based on density and detecting different data points.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Adzkia Nur Nasution
Universitas Negeri Medan
Email: adzkiaaanur41@gmai.com

INTRODUCTION

Advances in technology and science have facilitated the development of data analysis, making it increasingly important in various fields. Recent research considers data mining by using a variety of methods to find patterns and structures in data, such as clustering and predictive techniques based on specific parameters. In this context, data clustering plays an important role in the identification of flower species, thus allowing us to understand morphological variations and relationships between species. The beauty and uniqueness of flowers such as their color, aroma, and shape attract the attention of many people. However, due to the many variations, it can be difficult to recognize the name of the flower at first glance. One of the most prominent datasets in the study is the iris flower dataset from the UCI Machine Learning Repository.It includes three species: Setosa, Versicolor and Virginica, and attributes such as petal length and width. By applying clustering techniques to this dataset, we can more easily identify flower types and improve our understanding of the diversity of the plant kingdom. [1] [2]

Clustering is the process of grouping objects based on their similarity to each specific array partition. The purpose of cluster analysis is to group objects or individuals into groups based on their characteristics. This means that each group has internal similarities but is different from other groups. This analysis process

ISSN: 2615-8019

includes data standardization, object similarity measurement, and clustering method selection. Clustering is an important technique in data mining that helps uncover hidden patterns and structures in the data. An example of its application is the classification of flower types. This helps scientists distinguish and understand the similarities between species based on their characteristics. The main challenge in this process is the presence of noise or data that does not match the general pattern, which can reduce the accuracy of group decision-making. [3]

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a clustering technique that focuses on data density, called epsilon (ε), which is the smallest amount of data present within a given radius. This method collects data based on two main parameters: epsilon and minpts, which determine the number of clusters formed. The advantages of DBSCAN are that it can identify noisy data, identify parts of the cluster, and detect irrelevant data. Unlike other algorithms such as K-means, DBSCAN does not require a predetermined number of clusters and can detect outlier points well. Therefore, DBSCAN is suitable for application to irregular real data. [4] [5]

This study aims to apply the DBSCAN algorithm to the iris flower dataset by using Google Colab as a programming platform. Using this approach, this study analyzes the results of grouping and identifies data points that are classified as outliers. The expected results of this study not only provide a deeper understanding of the data structure of iris flowers, but also show how the DBSCAN algorithm can be used for more comprehensive and complex data analysis, as well as useful in the fields of ecology and biology.

METHOD

2.1. Data Collection

The data used in this study was obtained from the UCI Machine Learning Repository in the form of Iris Dataset. This dataset contains information on three species of flowers, namely Setosa, Versicolor, and Virginica, which are widely used as standards in classification and grouping studies. This dataset consists of 150 samples with a total of 5 attributes, namely Sepal Length (sepal length), Sepal Width (sepal width), Petal Length (petal length), Petal Width (petal width), and Species that indicate the type of species as a label. [6]

This dataset was analyzed using the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm, which is highly effective in handling spatial data due to its ability to manage irregular data as well as detect irrelevant or noise-containing data. This DBSCAN method allows grouping without specifying the number of clusters at the beginning, making it suitable for detecting the distribution patterns of three species in a dataset. [7]

2.2. DBSCAN Clustering

DBSCAN is a grouping algorithm capable of identifying high-density areas to form clusters. Clusters in this algorithm are defined as regions that have a dense or high-density concentration of samples, which are then distinguished from regions with low density (noise). The DBSCAN algorithm is a non-parametric algorithm in unsupervised learning, which means it does not rely on certain assumptions when grouping data. In principle, DBSCAN is able to form clusters in a flexible or unlimited form and effectively handle noise or outliers in the cluster. The algorithm identifies high-density areas as clusters using two main parameters that need to be carefully defined. Those parameters are the boundary radius, denoted as ε, and the minimum number of objects, denoted by MinObj, which are needed to determine whether an area belongs to a cluster. [8] [9]

2.3. DBSCAN Clustering Stages

In this study, the application of DBSCAN Clustering is carried out through several main stages that aim to obtain optimal data clusters. This stage includes data frame exploration, data preprocessing, normalization, model turning, model building and visualization model.

2.3.1. Exploring Data Frames

This stage includes an initial exploration of the structure and distribution of data on the Iris dataset. Data exploration is done by understanding variable distributions and early outlier detection, which is important in identifying data characteristics before the clustering process. In-depth exploration of the data can help understand the initial patterns that may emerge and provide insight into the potential cluster shapes that the DBSCAN algorithm will form.

2.3.2. Data Preprocessing: Normalization

This stage is the process of filtering the data before processing to obtain uniform attributes, so that unrepresentative data can be eliminated. Since each attribute has a different range of values, pre-processing is done in the form of normalization, which ensures all data on each attribute is in the same range before testing. In addition, the preprocess also removes incomplete data and organizes the data in a format appropriate for spatial analysis, adapting it to the needs of the analysis using multiple variables. [10] [11]

2.3.3. Model Turning

The main parameters of DBSCAN are the limit radius (ε) and the minimum number of objects (MinObj), which need to be carefully determined as these parameters affect DBSCAN's sensitivity in forming clusters and detecting noise. The ε value can be determined using the k-distance graph method, while the MinObj value is usually determined based on the dimensions of the dataset. Precise parameter determination will allow DBSCAN to optimally segment data without ignoring relevant noise.

2.3.4. Model Building

Once the parameters are defined, the DBSCAN algorithm is applied to the Iris dataset to form clusters based on the data density. At this stage, DBSCAN will identify the data point as a core point, border point, or noise based on the predefined parameters. The result of this process is the formation of clusters and the identification of points that are considered noise.

2.3.5. Model Visualization

Model visualization is carried out to display the results of the grouping that has been formed by DBSCAN. In this visualization, the resulting cluster and noise data can be seen clearly, providing an overview of the distribution of the clusters formed as well as the existence of data points that are considered noise. Cluster visualization is usually done using scatter plots or 3D plots to display the relationships between variables in the data. This visualization helps in evaluating the results and effectiveness of DBSCAN in forming representative clusters.

RESULTS AND DISCUSSION

This study was conducted to identify and analyze different or outlier data from groups of Iris flower species using the DBSCAN grouping method. To identify Iris flowers such as Iris setosa, Iris versicolor, and Iris virginica, we need to mark data that are different from the others (outliers). These outliers suggest that some specimens may have unusual characteristics compared to other specimens, either due to environmental influences, genetic variation, or measurement errors. Here are the steps in data processing.

3.1. Exploratory Data Frame

Data is processed using Google Colab. Data from Uci Mechine Learning was first analyzed by entering data into the system to display a summary of existing data. The Exploratory Data Frame is shown in figure 1.

	sepal length in cm	sepal width in cm	petal length in cm	petal width in cm	class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...
142	6.7	3.0	5.2	2.3	Iris-virginica
143	6.3	2.5	5.0	1.9	Iris-virginica
144	6.5	3.0	5.2	2.0	Iris-virginica
145	6.2	3.4	5.4	2.3	Iris-virginica
146	5.9	3.0	5.1	1.8	Iris-virginica

147 rows × 5 columns

Figure 1. Exploratory Data Frame

3.2. Data Pre-Processing: Normalizaton

Data pre-processing is the step to transform raw data into easy-to-understand data. Normalization is the process of making data have a value of zero and a standard deviation of one. Figure 2 shows that the use of the 'StandardScaler' code is used to standardize the data in the Data Frame.

```
from sklearn.preprocessing import StandardScaler

# kolom kategori bernama 'class'
data_numeric = data.drop(columns=[' class'])  # Drop kolom kategori
scaler = StandardScaler()
data_normalized = scaler.fit_transform(data_numeric)

# Tambahkan kembali kolom kategori ke data yang sudah dinormalisasi
data_normalized = pd.DataFrame(data_normalized, columns=data_numeric.columns)
data_normalized[' class'] = data[' class'].values  # Tambahkan kembali kolom kategori
```

Figure 2. Data Pre-processing: Normalization

ISSN: 2615-8019

3.3. Model Turning

The Turning model is a DBSCAN model which is an important step because DBSCAN works based on the grouping of data points based on the surrounding density. DBSCAN groups data in a different way than other commonly used methods. DBSCAN focuses on the number of data points in a given range that is sufficient to form a cluster, as opposed to methods that use geometric division. In this process, DBSCAN uses two main parameters, namely eps (epsilon or radius) and min_samples (the minimum number of dots in the radius that are considered the nucleus of the cluster). These two parameters have a direct effect on the final result of the grouping process. They determine how well the model can identify tight-knit groups and ignore data that is perceived as anomalies or noise. This is a complete explanation of the benefits of specifying each of these parameters in DBSCAN.

```
# Mengatur parameter DBSCAN
dbscan = DBSCAN(eps=0.5, min_samples=5)
clusters = dbscan.
```

ndarray: clusters

ndarray with shape (300,)

```
# Menambahkan kolo
data['cluster'] = clusters

# Menampilkan hasil clustering
print(data[[' class', 'cluster']].head())
```

Figure 3. Turning Model

3.4. Model Building

The Bulidilng model involves a different approach than traditional grouping techniques. DBSCAN is designed to search for groups based on the density of data points in space. In other words, the model does not calculate data based on average distances or mathematical equations, but rather based on density around a specific point. When developing the model using DBSCAN, the authors need to pay attention to two main parameters. The first is eps (epsilon) which specifies the maximum distance to connect data points. The second is min_samples that indicates the minimum number of points within a given radius for a single point to be considered part of a cluster. By using the DBSCAN method in model building, we can perform more accurate data grouping for complex data such as geospatial data, collection points in medical images, and object classification based on density. This process ensures that the model not only generates meaningful clusters but also identifies data that does not fit into the usual patterns, thus providing a more complete and detailed analysis of the data.

```
# Menggunakan hanya fitur numerik untuk clustering
X = data.iloc[:, :-1].values

# Standarisasi data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Menampilkan data yang sudah distandarisasi
print(X_scaled[:5])
```

```
[[-0.90068117  1.01900435 -1.34022653 -1.3154443 ]
 [-1.14301691 -0.13197948 -1.34022653 -1.3154443 ]
 [-1.38535265  0.32841405 -1.39706395 -1.3154443 ]
 [-1.50652052  0.09821729 -1.2833891  -1.3154443 ]
 [-1.02184904  1.24920112 -1.34022653 -1.3154443 ]]
```

Figure 4. Model Building

3.5. Model Visualization

This visualization model displays the outlier numbers in the Iris data. Based on the visualization of cluster segmentation, the object points are grouped into three clusters: cluster -1 (blue), cluster 0 (orange), and cluster 1 (green) and the dot shapes show the respective species of data A black circle, a black square for versicolor and a black rhombus shape for virginica. The blue dot is the data that is considered an outlier by the algorithm. In the DBSCAN method used in this visualization, groups are formed based on data density. The DBSCAN method identifies groups based on the distance between the data from each other that is less than a certain value (epsilon) and the minimum number of data points within that distance. This visualization shows that the DBSCAN method can group data by density and detect points that do not belong to the main group (in blue), referred to as outliers.

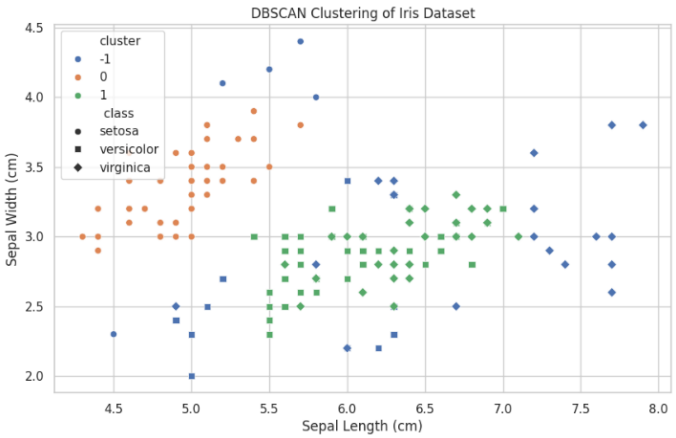

Figure 5. Model Visualization

3.6. Results and Evaluation

The results of the grouping showed that the data was divided into three groups, namely groups -1, 0, and 1. Based on the amount of data available, cluster 1 has 71 data, cluster 0 has 45 data, and cluster -1 which is considered an outlier has 34 data. These groups can be further identified based on the species of flowers in the Iris dataset.

The details of the clusters by flower species are as follows:

a. Cluster -1 consists of 5 setosa data, 11 versicolor data, and 18 virginica data.
b. Cluster 0 is entirely composed of setosa species with a total of 45 data.
c. Cluster 1 consists of 39 versicolor data and 32 virginica data.

From these results, it can be concluded that the DBSCAN method successfully separates several groups based on flower species. Setosa species are well classified into a single group (cluster 0), while the species versicolor and virginica form separate groups (cluster 1). Some data from these three species were identified as outliers (group -1). This suggests that DBSCAN can identify dense data groups and find outliers that do not belong to the main group.

```
cluster
 1    71
 0    45
-1    34
Name: count, dtype: int64
cluster     -1   0   1
 class
setosa       5  45   0
versicolor  11   0  39
virginica   18   0  32
Terdapat beberapa cluster yang terpisah berdasarkan spesies bunga.
```

Figure 6. Results and Evaluation

CONCLUSION

This study successfully showed that the DBSCAN method is effective in grouping iris flower species data based on density and detecting different data points (outliers). The grouping results divided the data into three main clusters, where cluster 0 consisted entirely of Setosa species, while cluster 1 included Versicolor and Virginica species. The -1 cluster represents data identified as outliers, indicating some specimens with unusual characteristics compared to others. The DBSCAN application has been shown to be able to separate species based on similarity in traits and detect data that differ from common patterns, making it a powerful method for complex and irregular data analysis.

Suggestion

For further research, it is recommended to apply the DBSCAN algorithm to larger or more complex datasets to test its effectiveness in detecting outliers in high-dimensional data. In addition, comparisons with other grouping methods can provide deeper insight into the accuracy and advantages of DBSCAN in species classification and outlier detection, thus providing more comprehensive results.

ISSN: 2615-8019

REFERENCES

[1] N. N. Hasanah and A. S. Purnomo, "Implementation of Data Mining for Book Grouping Using K-Means Clustering Algorithm (Case Study: LPP Yogyakarta Polytechnic Library)," Journal of Business Information Technology and Systems, vol. 4, no. 2, pp. 300–311, 2022.

[2] D. T. Worung, S. R. U. A. Sompie, and A. Jacobus, "Implementation of K-Means and K-NN in the Classification of Flower Images," Journal of Computer Engineering, vol. 15, no. 3, pp. 217–222, 2021.

[3] P. R. N. Saputra, A. Chusyairi, and others, "Comparison of clustering methods in the grouping of puskesmas data on complete basic immunization coverage," RESTI Journal (Systems Engineering and Information Technology), vol. 4, no. 6, pp. 1077–1084, 2020.

[4] E. K. Sihite, Y. M. Rangkuti, and I. K. Karo, "Webgis Development for Malnutrition Sufferers in Medan City Based on the Results of DBSCAN Algorithm Clustering," SAINTIKOM Journal (Journal of Informatics and Computer Management Science), vol. 23, no. 1, pp. 77–86, 2024.

[5] J. Riyono, C. E. Pujiastuti, and A. L. R. Putri, "COUNTRY CLUSTERING BASED ON TOBACCO CONSUMPTION CONTROL SCORE USING DBSCAN ALGORITHM," JTIK (Journal of Computer Engineering), vol. 8, no. 1, pp. 78–89, 2024.

[6] S. Aldana and J. S. Wibowo, "The Application of Data Mining to the Classification of Patients with Liver Disease Using the K-Nearest Neighbor Method," Progressive: A Journal of Computer Science, vol. 20, no. 1, pp. 124–132, 2024.

[7] R. Fadhlillah et al., "Analysis of Population Density Clustering in Muara Enim Regency Using DBSCAN Algorithm," in PROCEEDINGS OF THE NATIONAL SEMINAR ON DATA SCIENCE, 2024, pp. 982–992.

[8] D. P. Indini, S. R. Siburian, N. Nurhasanah, D. P. Utomo, and M. Mesran, "Implementation of DBSCAN Algorithm for Clustering Selection of Students Eligible to Receive Foundation Scholarships," ESCAF, pp. 1328–1335, 2022.

[9] A. Saputra and R. Yusuf, "Comparison of the DBSCAN and K-MEANS Algorithms in Segmenting Customers Using Public Transportation of Transjakarta Using the RFM Method," MALCOM: Indonesian Journal of Machine Learning and Computer Science, vol. 4, no. 4, pp. 1346–1361, 2024.

[10] A. Homaidi, A. Lutfi, and others, "Implementation of Clustering Method with DBSCAN Algorithm for Identification of Industrial Centers Based on Google Map," G-Tech: Journal of Applied Technology, vol. 8, no. 3, pp. 2112–2121, 2024.

[11] A. S. Ritonga and I. Muhandhis, "Data Mining Techniques to Classify Tourist Destination Review Data Using Principal Component Analysis (Pca) Data Reduction," Edutic Scientific Journal: Education and Informatics, vol. 7, no. 2, pp. 124–133, 2021.