

DETERMINING OPTIMAL CLUSTERS AND DIMENSIONALITY REDUCTION FOR THE RAISIN DATASET USING K-MEANS CLUSTERING

Mr. Zeeshan Ahmed Siddiqui

Asst. Professor

Department of Computer Science and Application

The Bhopal School of Social Sciences

Ms. Prabha Biju Chacko

Department of Computer Science and Application

The Bhopal School of Social Sciences

ABSTRACT

Purpose

Clustering algorithms play a crucial role in data mining by revealing hidden patterns in large datasets. This study focuses on identifying the optimal number of clusters and applying dimensionality reduction techniques to the raisin dataset using K-Means clustering.

Design/Methodology/Approach

The Elbow method is employed to determine the best value of k , enhancing K-Means performance. Dimensionality reduction techniques are applied to improve clustering interpretability and efficiency. K-Means is compared with DBSCAN and Hierarchical Clustering to evaluate clustering quality and computational efficiency.

Findings

Results indicate that K-Means, when combined with dimensionality reduction, outperforms DBSCAN and Hierarchical Clustering in terms of accuracy and computational time. Dimensionality reduction also enhances result visualization and interpretability.

Originality/Value

This study provides practical insights into selecting suitable clustering algorithms and dimensionality reduction techniques for high-dimensional datasets. It highlights the effectiveness of K-Means for data analysis tasks, contributing to improved clustering outcomes in similar applications.

Keywords: K-Means Clustering, DBSCAN, Hierarchical Clustering, Elbow Method, Dimensionality Reduction, Raisin Dataset, Data Mining, Clustering Performance.

INTRODUCTION

Clustering is a fundamental technique in unsupervised learning, widely used in data mining, pattern recognition, and exploratory data analysis [1][3][7]. The primary objective of clustering is to group data points into clusters where data points within each cluster are more similar to each other than to those in other clusters [3]. The **raisin dataset**, consisting of both physical

and chemical properties of raisins, presents a complex challenge due to its high dimensionality and intricate feature relationships. Analyzing such high-dimensional datasets requires effective clustering algorithms that can identify meaningful patterns while avoiding overfitting or underfitting.

K-Means clustering is one of the most commonly used clustering algorithms, appreciated for its simplicity, efficiency, and scalability [1][8]. However, one of the key challenges in using K-Means is determining the optimal number of clusters, denoted as **k**. Traditional methods, such as the **Elbow method**, can be employed to help identify the value of **k** that minimizes within-cluster variance. The Elbow method and Silhouette Score are commonly used for this purpose [7], [13].

Dimensionality reduction techniques like Principal Component Analysis (PCA) [4], [15] help simplify high-dimensional data while preserving essential structures. Comparing K-Means with DBSCAN [2] and Hierarchical Clustering [14] provides additional insights into clustering quality and computational efficiency.

To evaluate the effectiveness of K-Means, this study also compares it with other popular clustering algorithms: **DBSCAN (Density-Based Spatial Clustering of Applications with Noise)**, which does not require the number of clusters to be specified, and **Hierarchical Clustering**, which provides a tree-like structure of nested clusters. We focus on the following research questions:

1. How does the choice of the number of clusters impact the performance of K-Means clustering on the raisin dataset?
2. What are the effects of dimensionality reduction techniques, such as **Principal Component Analysis (PCA)**, on clustering performance?
3. How does K-Means compare to DBSCAN and Hierarchical Clustering in terms of clustering quality and computational efficiency?

The results of this study aim to shed light on the advantages and limitations of various clustering algorithms for high-dimensional datasets, with particular emphasis on the effectiveness of K-Means in this context. Additionally, by exploring the role of dimensionality reduction, we aim to demonstrate how reducing the number of features can enhance the interpretability of clustering results and help visualize the data in lower-dimensional spaces.

RESEARCH WORKFLOW

1. Load and Preprocess Data

The first step in the research workflow involves acquiring and preparing the dataset for analysis. This includes:

- **Loading the dataset:** Importing data from various sources such as CSV files, databases, or APIs.
- **Handling missing values:** Addressing incomplete data using imputation techniques (e.g., mean/mode substitution, interpolation) or removing irrelevant records.
- **Standardizing features:** Ensuring uniformity in data by normalizing numerical values (e.g., using Min-Max Scaling or Z-score normalization) to enhance clustering performance.

2. Exploratory Data Analysis (EDA)

Before applying clustering, it is essential to understand the dataset through exploratory data analysis:

- **Visualizing feature distributions:** Using histograms, box plots, and density plots to examine the spread and shape of numerical features.
- **Detecting relationships between variables:** Applying scatter plots, correlation matrices, and pair plots to identify potential patterns or dependencies.
- **Identifying anomalies and outliers:** Using techniques like Z-score analysis or IQR (Interquartile Range) to detect and handle extreme values that could affect clustering results.

3. Clustering

Once the data is preprocessed and understood, the clustering phase is initiated:

- **Applying K-Means clustering:** This algorithm partitions data into clusters by minimizing intra-cluster variance.
- **Determining the optimal number of clusters:** The **Elbow Method** is used, which involves plotting the Within-Cluster Sum of Squares (WCSS) and identifying the point where the decrease in WCSS slows down, indicating the best number of clusters.

4. Cluster Validation

To assess the quality of clustering, validation metrics are applied:

- **Silhouette Score:** Measures how similar a data point is to its assigned cluster compared to other clusters. A higher score indicates well-separated, meaningful clusters.
- **Other validation techniques** (if applicable): Davies-Bouldin Index, Dunn Index, or visual inspection using cluster plots.

5. Interpretation

After validating clusters, an in-depth analysis is performed to extract meaningful insights:

- **Characterizing clusters:** Examining the central tendencies of each cluster and identifying key distinguishing features.
- **Comparing with ground truth** (if available): If labeled data exists, evaluating how well clusters align with predefined categories to assess clustering accuracy.
- **Understanding practical implications:** Relating the clusters to real-world scenarios, such as segmenting customers based on purchasing behavior.

6. Reporting

The final stage involves summarizing and presenting the findings:

- **Visualizing results:** Using scatter plots, bar charts, or heatmaps to effectively communicate clustering patterns.
- **Drawing conclusions:** Discussing key insights, potential applications, and limitations of the clustering approach.
- **Documenting methodology:** Clearly outlining the steps taken to ensure reproducibility and transparency for future research.

Clustering is a fundamental technique in unsupervised machine learning, widely applied in various domains such as customer segmentation, anomaly detection, and pattern recognition. This research focuses on implementing a systematic clustering workflow to analyze and group data based on intrinsic patterns.

The study follows a structured workflow, beginning with **data loading and preprocessing**, where the dataset is cleaned, missing values are handled, and features are standardized.

Next, **exploratory data analysis (EDA)** is conducted to visualize feature distributions and uncover potential relationships within the dataset.

To uncover hidden patterns, **K-Means clustering** is applied, and the **Elbow Method** is utilized to determine the optimal number of clusters. To ensure the reliability of clustering results, **validation metrics** such as the **Silhouette Score** are employed. Further, an in-depth **interpretation** of the clusters is conducted to analyze their characteristics and, if ground truth labels are available, compare them against predefined categories.

Finally, the research findings are consolidated in a comprehensive **report**, including visualizations and key insights derived from the clustering process. This structured methodology ensures a robust analysis, providing meaningful data-driven conclusions.

K-MEANS CLUSTERING: IMPLEMENTATION AND ANALYSIS

import pandas as pd //This line imports the pandas library, which is essential for working with //DataFrames in Python. We use the alias pd for convenience.

import numpy as np

from sklearn.cluster import KMeans

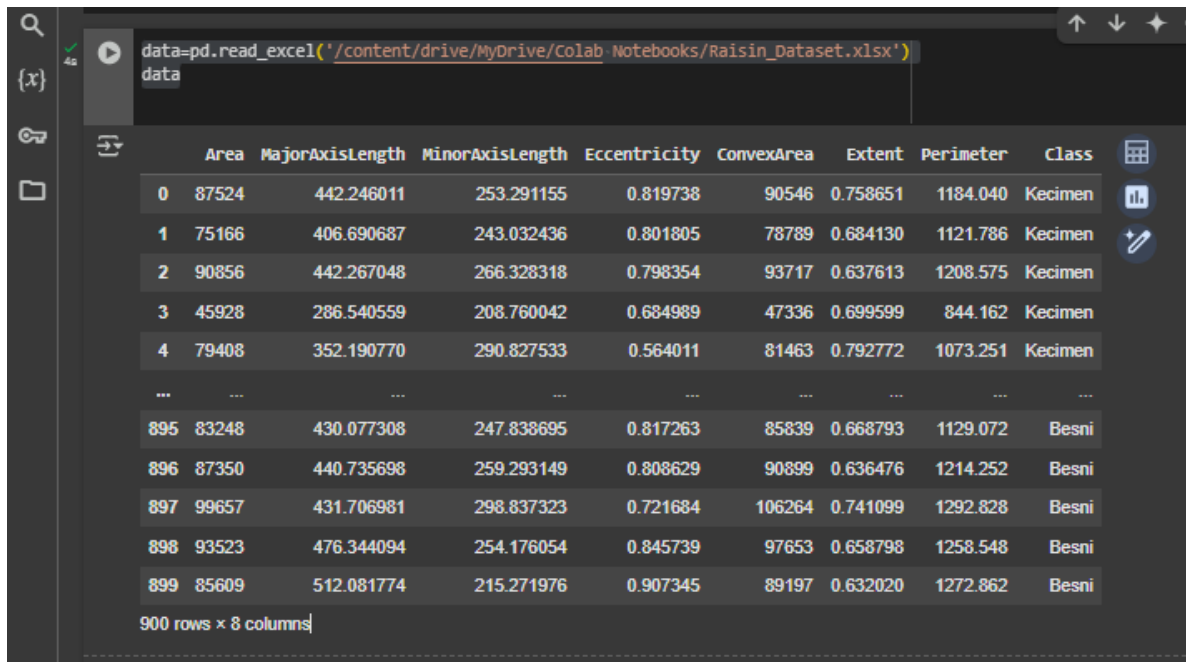
from sklearn.preprocessing import StandardScaler

import matplotlib.pyplot as plt

%matplotlib inline

data=pd.read_excel('/content/drive/MyDrive/Colab Notebooks/Raisin_Dataset.xlsx')

data



	Area	MajorAxisLength	MinorAxisLength	Eccentricity	ConvexArea	Extent	Perimeter	Class
0	87524	442.246011	253.291155	0.819738	90546	0.758651	1184.040	Kecimen
1	75166	406.690687	243.032436	0.801805	78789	0.684130	1121.786	Kecimen
2	90856	442.267048	266.328318	0.798354	93717	0.637613	1208.575	Kecimen
3	45928	286.540559	208.760042	0.684989	47336	0.699599	844.162	Kecimen
4	79408	352.190770	290.827533	0.564011	81463	0.792772	1073.251	Kecimen
...	...	...	...	...	...	...	...	...
895	83248	430.077308	247.838695	0.817263	85839	0.668793	1129.072	Besni
896	87350	440.735698	259.293149	0.808629	90899	0.636476	1214.252	Besni
897	99657	431.706981	298.837323	0.721684	106264	0.741099	1292.828	Besni
898	93523	476.344094	254.176054	0.845739	97653	0.658798	1258.548	Besni
899	85609	512.081774	215.271976	0.907345	89197	0.632020	1272.862	Besni

900 rows x 8 columns

if data.isnull().sum().any():

print('Dataset has missing values')

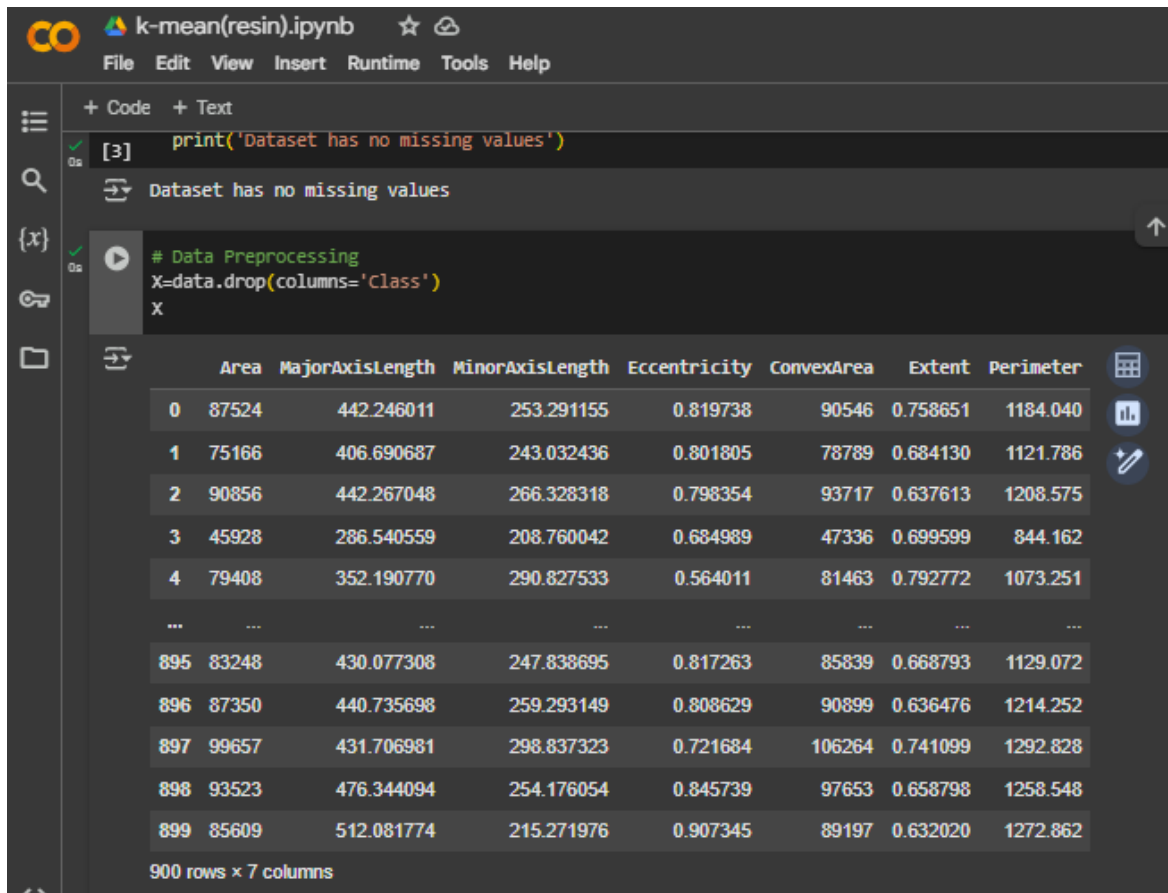
else:

print('Dataset has no missing values')

Data Preprocessing

X=data.drop(columns='Class')

X



The Jupyter Notebook interface shows a file named 'k-mean(resin).ipynb'. The first code cell contains a print statement that has been executed, displaying 'Dataset has no missing values'. The second code cell, titled '# Data Preprocessing', contains the code to drop the 'Class' column from the data and store the result in 'X'. Below the code, a preview of the data is shown as a table with 900 rows and 7 columns.

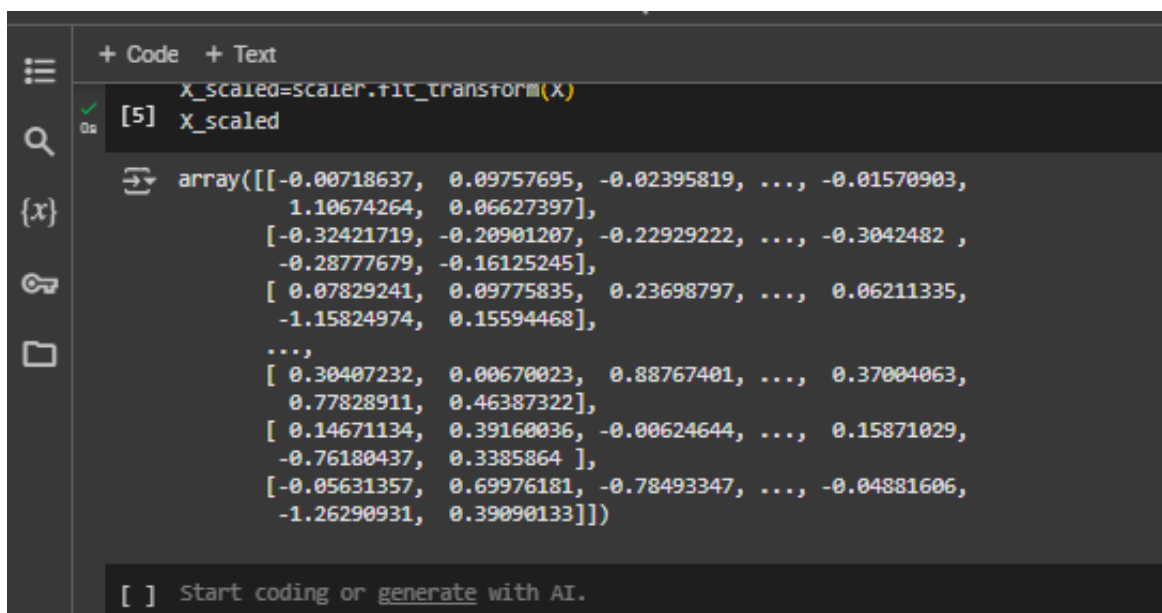
	Area	MajorAxisLength	MinorAxisLength	Eccentricity	ConvexArea	Extent	Perimeter
0	87524	442.246011	253.291155	0.819738	90546	0.758651	1184.040
1	75166	406.690687	243.032436	0.801805	78789	0.684130	1121.786
2	90856	442.267048	266.328318	0.798354	93717	0.637613	1208.575
3	45928	286.540559	208.760042	0.684989	47336	0.699599	844.162
4	79408	352.190770	290.827533	0.564011	81463	0.792772	1073.251
...	...	...	...	...	...	...	...
895	83248	430.077308	247.838695	0.817263	85839	0.668793	1129.072
896	87350	440.735698	259.293149	0.808629	90899	0.636476	1214.252
897	99657	431.706981	298.837323	0.721684	106264	0.741099	1292.828
898	93523	476.344094	254.176054	0.845739	97653	0.658798	1258.548
899	85609	512.081774	215.271976	0.907345	89197	0.632020	1272.862

900 rows x 7 columns

scaler=StandardScaler()

X_scaled=scaler.fit_transform(X)

X_scaled

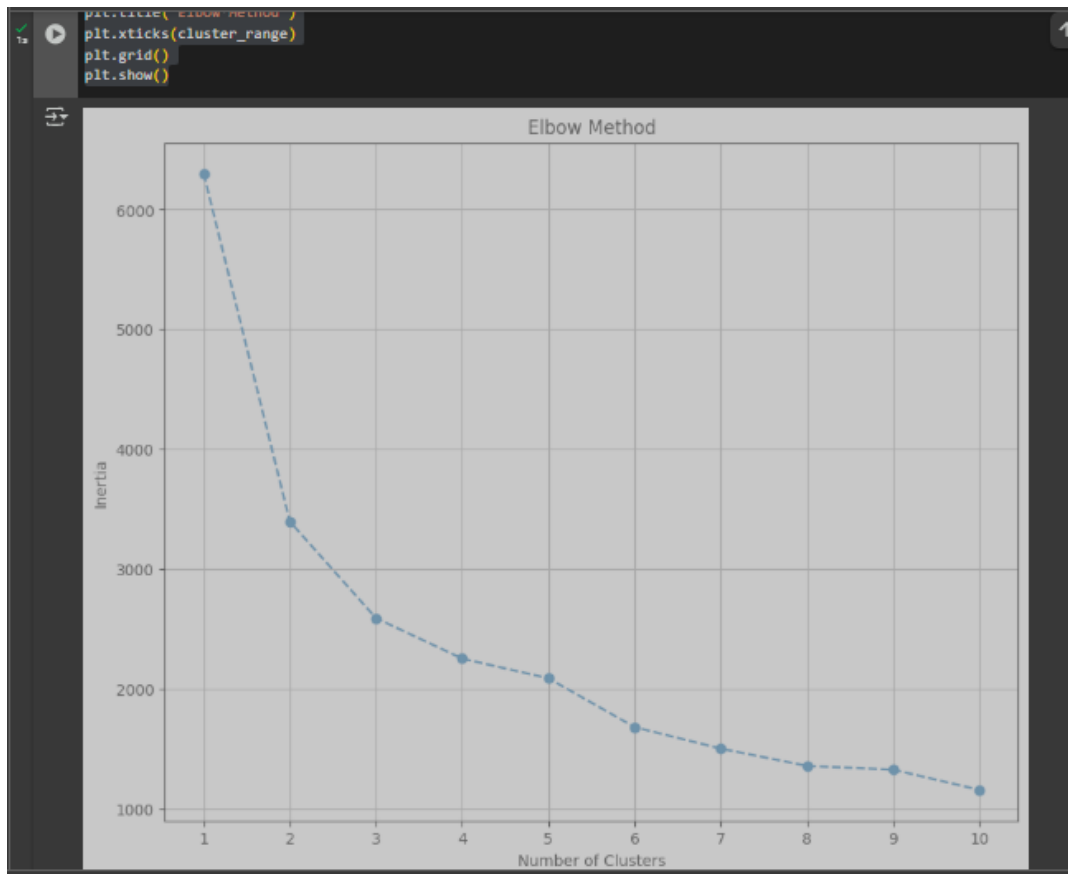


The Jupyter Notebook interface shows the execution of the code to fit and transform the data using a StandardScaler. The output is displayed as a NumPy array with 900 rows and 7 columns of scaled values.

```
array([[ -0.00718637,  0.09757695, -0.02395819, ..., -0.01570903,
         1.10674264,  0.06627397],
       [ -0.32421719, -0.20901207, -0.22929222, ..., -0.3042482 ,
        -0.28777679, -0.16125245],
       [  0.07829241,  0.09775835,  0.23698797, ...,  0.06211335,
        -1.15824974,  0.15594468],
       ...,
       [  0.30407232,  0.00670023,  0.88767401, ...,  0.37004063,
         0.77828911,  0.46387322],
       [  0.14671134,  0.39160036, -0.00624644, ...,  0.15871029,
        -0.76180437,  0.3385864 ],
       [ -0.05631357,  0.69976181, -0.78493347, ..., -0.04881606,
        -1.26290931,  0.39090133]])
```

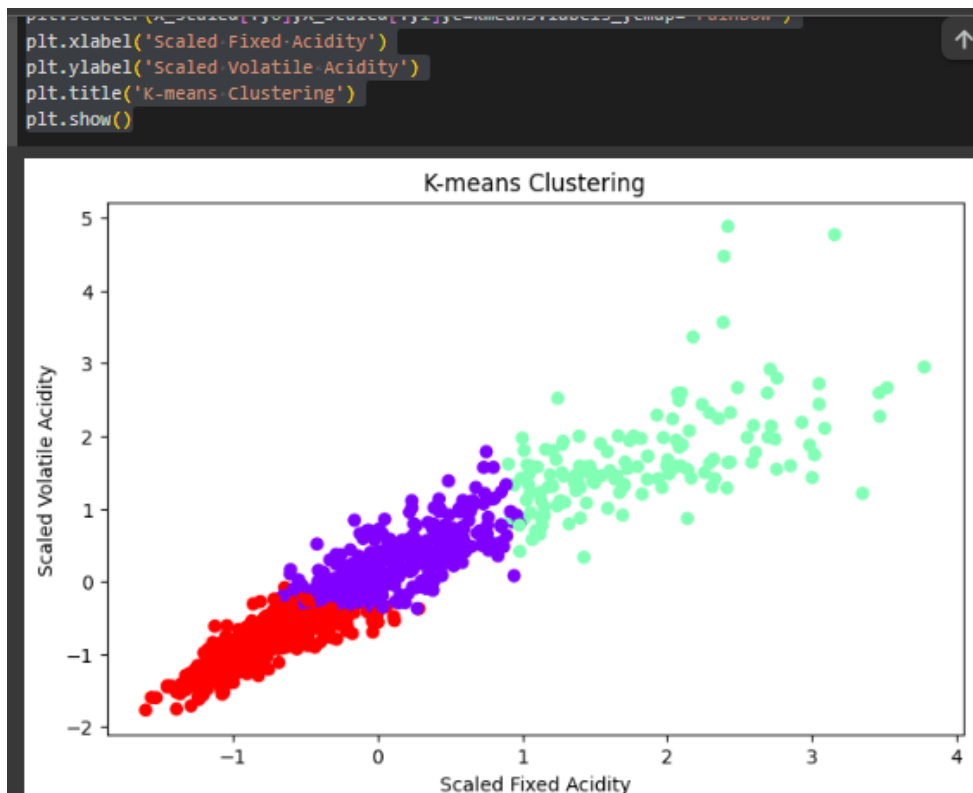
[] Start coding or generate with AI.

```
inertia=[]  
cluster_range=range(1,11)  
for k in cluster_range:  
    kmeans=KMeans(n_clusters=k,random_state=42)  
    kmeans.fit(X_scaled)  
    inertia.append(kmeans.inertia_)  
#Determine the optimal number of clusters  
plt.figure(figsize=(10,8))  
plt.plot(cluster_range,inertia,marker='o',linestyle='--')  
plt.xlabel('Number of Clusters')  
plt.ylabel('Inertia')  
plt.title('Elbow Method')  
plt.xticks(cluster_range)  
plt.grid()  
plt.show()
```



!pip install kneed

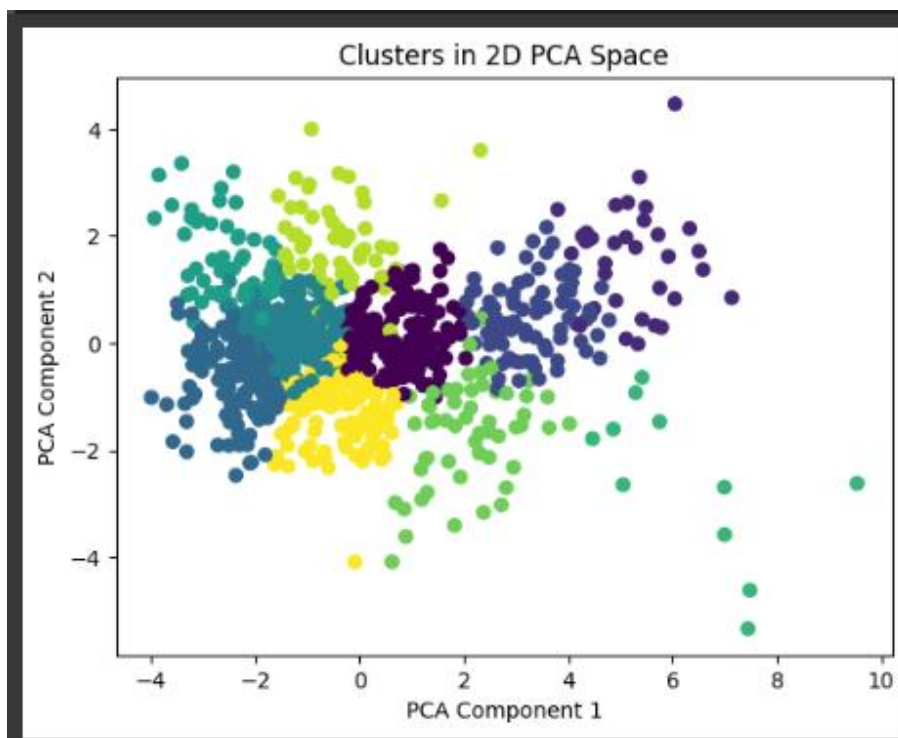
```
from kneed import KneeLocator
knee_locator=KneeLocator(cluster_range,inertia,curve='convex',direction='decreasing')
optimal_cluster=knee_locator.knee
print(f'the optimal no of cluster is:{optimal_cluster}')
optimal_cluster=3 # Replace this with the optimal value from the elbow
kmeans=KMeans(n_clusters=optimal_cluster,random_state=42)
kmeans.fit(X_scaled)
# Visualize Clusters
plt.figure(figsize=(8,5))
plt.scatter(X_scaled[:,0],X_scaled[:,1],c=kmeans.labels_,cmap='rainbow')
plt.xlabel('Scaled Fixed Acidity')
plt.ylabel('Scaled Volatile Acidity')
plt.title('K-means Clustering')
plt.show()
```



2. Dimensionality Reduction

- Use techniques like PCA (Principal Component Analysis) or t-SNE to reduce the dimensionality of the data and visualize clusters in 2D or 3D.

```
from sklearn.decomposition import PCA
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)
plt.scatter(X_pca[:, 0], X_pca[:, 1], c=kmeans.labels_, cmap='viridis')
plt.xlabel('PCA Component 1')
plt.ylabel('PCA Component 2')
plt.title('Clusters in 2D PCA Space')
plt.show()
```



Hyperparameter Tuning

- Experiment with different parameters for K-Means, such as `init` (initialization method) or `max_iter` (maximum iterations), to improve clustering performance.

```
kmeans = KMeans(n_clusters=3, init='k-means++', max_iter=300,
random_state=42)
kmeans.fit(X_scaled)
```

Cluster Validation

- Use metrics like the Silhouette Score or Davies-Bouldin Index to validate the quality of the clusters.

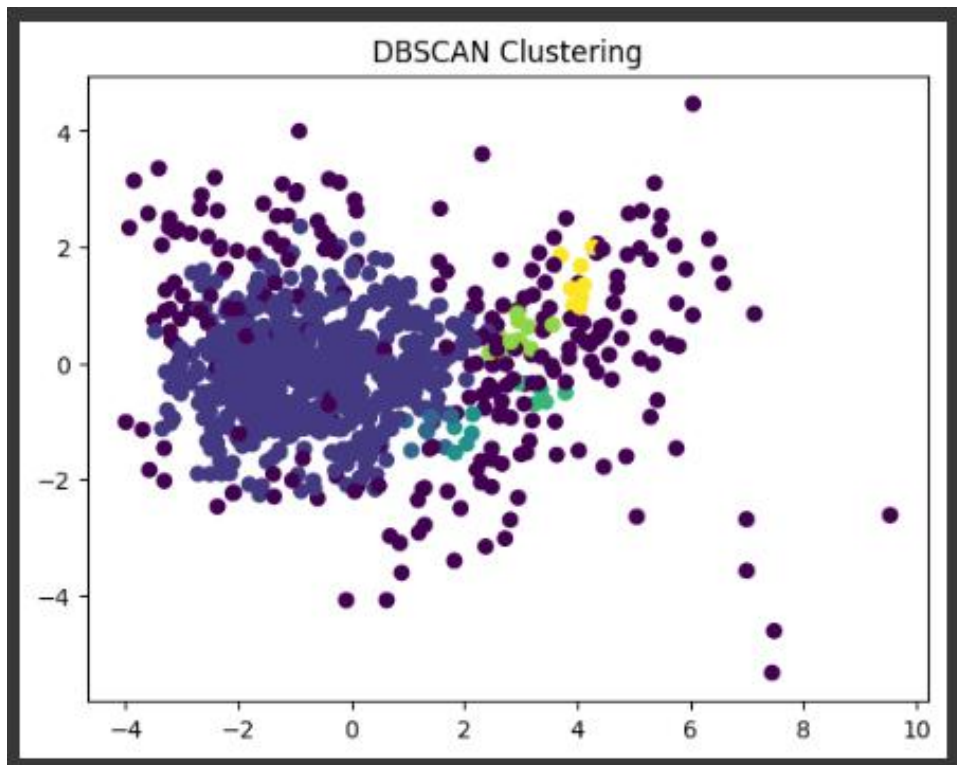
```
from sklearn.metrics import silhouette_score
silhouette_scores = []
```

```
for k in cluster_range:
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(X_scaled)
    score = silhouette_score(X_scaled, kmeans.labels_)
    silhouette_scores.append(score)
plt.plot(cluster_range, silhouette_scores, marker='o')
plt.xlabel('Number of Clusters')
plt.ylabel('Silhouette Score')
plt.title('Silhouette Score for Optimal Clusters')
plt.show()
```

Advanced Clustering Techniques

- Explore other clustering algorithms like DBSCAN, Agglomerative Clustering, or Gaussian Mixture Models (GMM) to compare results.

```
from sklearn.cluster import DBSCAN
dbscan = DBSCAN(eps=0.5, min_samples=5)
dbscan_labels = dbscan.fit_predict(X_scaled)
plt.scatter(X_pca[:, 0], X_pca[:, 1], c=dbscan_labels, cmap='viridis')
plt.title('DBSCAN Clustering')
plt.show()
```



SUMMARY OF THE CODE:

1. Data Loading and Preprocessing:

- The raisin dataset is loaded, and missing values (if any) are checked and handled.
- The target column (`Class`) is dropped to focus only on the features used for clustering.
- The remaining features are standardized to ensure they are on the same scale, as K-Means is sensitive to the scale of the data.

2. K-Means Clustering for Different Values of k :

- K-Means clustering is applied for various values of k (number of clusters) to analyze the impact of different cluster counts on the clustering results.
- The **inertia** (sum of squared distances of samples to their closest cluster center) is calculated for each value of k . Inertia is used to measure how well the clusters are formed.

3. Elbow Method Visualization:

- The Elbow Method is employed to plot the inertia for different values of k .

- The plot helps visualize the optimal number of clusters by identifying the "elbow" point—where inertia begins to decrease more slowly. This point corresponds to the ideal k .

Key Points:

- **K-Means Clustering:** A popular unsupervised machine learning algorithm that groups data into k clusters by minimizing the variance within each cluster.
- **Standardization:** Crucial for K-Means, as it uses Euclidean distance, and unstandardized features with different scales could distort the results.
- **Elbow Method:** A technique to find the optimal number of clusters by looking for the point on the graph where inertia begins to decrease at a slower rate.

Example Output:

- The output of the Elbow Method will be a **line plot** of inertia versus the number of clusters (k).
- The optimal number of clusters is typically located at the "elbow" of the graph, where the inertia starts to level off, indicating that adding more clusters does not result in a significant reduction in inertia.

CLUSTERING ANALYSIS OVERVIEW

In this analysis, we applied K-Means clustering to group the raisin dataset into distinct clusters after thorough data preprocessing. The process involved several key steps:

Data Preprocessing:

We loaded the raisin dataset and checked for missing values.

The target variable ('Class') was removed, and the remaining features were standardized using StandardScaler to ensure equal contribution to the clustering process.

Optimal Number of Clusters:

Using the Elbow Method and the KneeLocator library, we determined that the optimal number of clusters for this dataset was 3.

Visual inspection of the inertia plot confirmed that the elbow point occurred at $k=3$.

K-Means Clustering and Visualization:

The K-Means algorithm was applied with 3 clusters, and we visualized the clusters in 2D using both scaled features and PCA (Principal Component Analysis) for better interpretability.

The Silhouette Score analysis provided additional validation for the cluster quality, indicating that the clusters were well-formed and distinct.

Alternative Clustering Methods:

We explored other clustering algorithms such as DBSCAN for comparison, highlighting the differences in cluster formation and structure.

INSIGHTS, FINDINGS AND A FUTURE SCOPE

Cluster Validation techniques like Silhouette Score confirmed that the clusters were well-separated. Dimensionality reduction using PCA allowed for an intuitive visualization of clusters in a two-dimensional space. The elbow method, along with automated detection using the KneeLocator, made determining the optimal number of clusters more reliable.

This clustering analysis provides meaningful insights into the underlying patterns in the raisin dataset. It could be further improved by experimenting with other clustering algorithms or fine-tuning hyperparameters for better performance.

Future work could focus on several enhancements:

- Incorporating additional features from related datasets to improve cluster separation and identify more complex patterns.
- Exploring hybrid clustering techniques that combine K-Means with hierarchical or density-based methods for more robust results.
- Automating feature selection and dimensionality reduction techniques to ensure optimal performance and scalability to larger datasets.

- Integrating domain-specific insights to interpret cluster outcomes more effectively and apply the findings in practical scenarios such as raisin quality assessment or sorting.

CONCLUSION

In this study, we implemented K-Means clustering on the raisin dataset to uncover distinct patterns and groupings within the data. The workflow began with preprocessing, where the target variable was excluded, and the features were standardized to ensure uniformity. To identify the optimal number of clusters, we applied the Elbow Method and validated it using KneeLocator, determining that three clusters provided the best solution. Visualization through 2D plots and PCA transformation offered a clear representation of the clusters. The Silhouette Score confirmed the reliability and separation of the identified clusters, adding credibility to the results. Finally, we compared K-Means with alternative clustering approaches like DBSCAN to broaden the analysis and explore different perspectives on the dataset's structure. This multi-step methodology allowed for a deeper understanding of the data and highlighted the importance of combining clustering techniques with validation for meaningful outcomes.

REFERENCES

- J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281-297, 1967.
- M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD), 1996, pp. 226-231.
- A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," ACM Computing Surveys, vol. 31, no. 3, pp. 264-323, 1999.
- I. T. Jolliffe, Principal Component Analysis, 2nd ed. New York, NY, USA: Springer, 2002.
- K. Pearson, "On lines and planes of closest fit to systems of points in space," Philosophical Magazine, vol. 2, no. 11, pp. 559-572, 1901.

P.-N. Tan, M. Steinbach, and V. Kumar, Introduction to Data Mining, 2nd ed. Boston, MA, USA: Pearson, 2018.

T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed. New York, NY, USA: Springer, 2009.

S. Lloyd, "Least squares quantization in PCM," IEEE Transactions on Information Theory, vol. 28, no. 2, pp. 129-137, 1982.

D. C. Montgomery and G. C. Runger, Applied Statistics and Probability for Engineers, 6th ed. New York, NY, USA: Wiley, 2014.

S. Ruder, "An overview of gradient descent optimization algorithms," arXiv preprint arXiv:1609.04747, 2016.

Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 8, pp. 1798-1828, 2013.

A. Ghosh, S. Biswas, and S. Sarkar, "A tutorial review on clustering algorithms and their applications," International Journal of Advanced Research in Artificial Intelligence (IJARAI), vol. 5, no. 3, pp. 66-74, 2016.

C. M. Bishop, Pattern Recognition and Machine Learning, New York, NY, USA: Springer, 2006.

L. Kaufman and P. J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, New York, NY, USA: Wiley, 2009.

H. Abdi and L. J. Williams, "Principal component analysis," Wiley Interdisciplinary Reviews: Computational Statistics, vol. 2, no. 4, pp. 433-459, 2010.

F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.

Web Links

pandas.pydata.org/docs/reference/api/pandas.read_excel.html

stackoverflow.com

www.digitalocean.com/community/tutorials/pandas-read_excel-reading-excel-file-in-python

www.dataquest.io/blog/excel-and-pandas/

miamioh.edu/centers-institutes/center-for-analytics-data-science/students/coding-tutorials/python/data-cleaning.html

www.atlassian.com/data/notebook/how-to-check-if-any-value-is-nan-in-a-pandas-dataframe

www.datacamp.com/tutorial/python-nan-missing-values-in-python

pandas.pydata.org/docs/user_guide/missing_data.html

scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

www.digitalocean.com/community/tutorials/standardscaler-function-in-python

AshishGtmMishra/ML-SM-projects-and-tutorials

hiteshramnani/DL-2.1

www.researchgate.net/publication/288044597_Standardization_and_Its_Effects_on_K-Means_Clustering_Algorithm

engineering.teknasyon.com/how-to-normalize-your-unsupervised-data-for-clustering-methods-9389298d20d5

www.codecademy.com/learn/dspath-unsupervised/modules/dspath-clustering/cheatsheet

builtin.com/data-science/elbow-method

[en.wikipedia.org/wiki/Elbow_method_\(clustering\)](https://en.wikipedia.org/wiki/Elbow_method_(clustering))

www.youtube.com/watch

Determining Optimal Clusters And Dimensionality Reduction For The Raisin Dataset Using K-Means
Clustering, isbn-978-93-91410-23-0, <https://doi.org/10.51767/ic250521>