**Experiment: 9**

# PART A – DBSCAN

**Aim:** To study and analyse DBSCAN

**Software Used:** Jupyter Notebook.

## Theory:

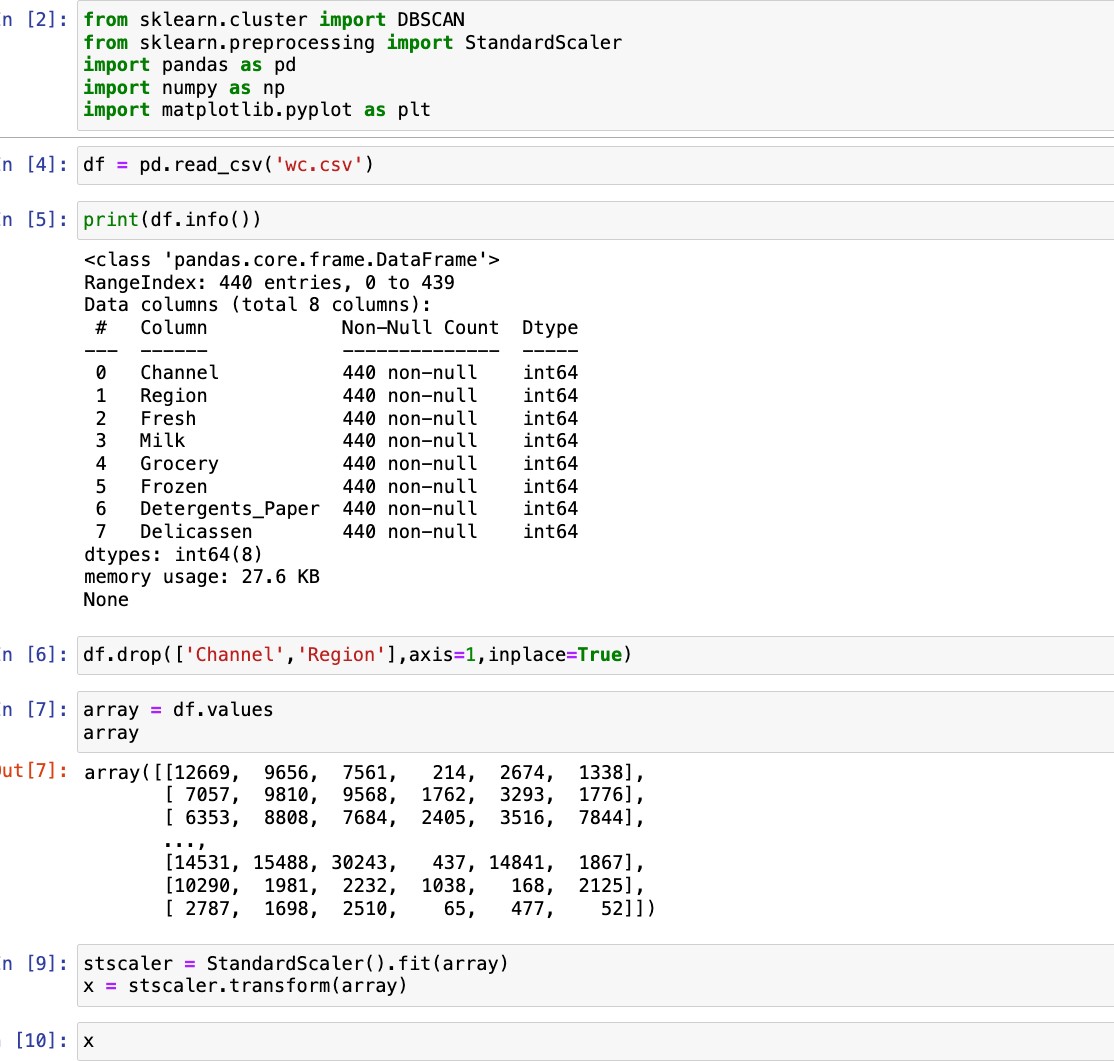
DBSCAN (Density Based Clustering of Applications with Noise):

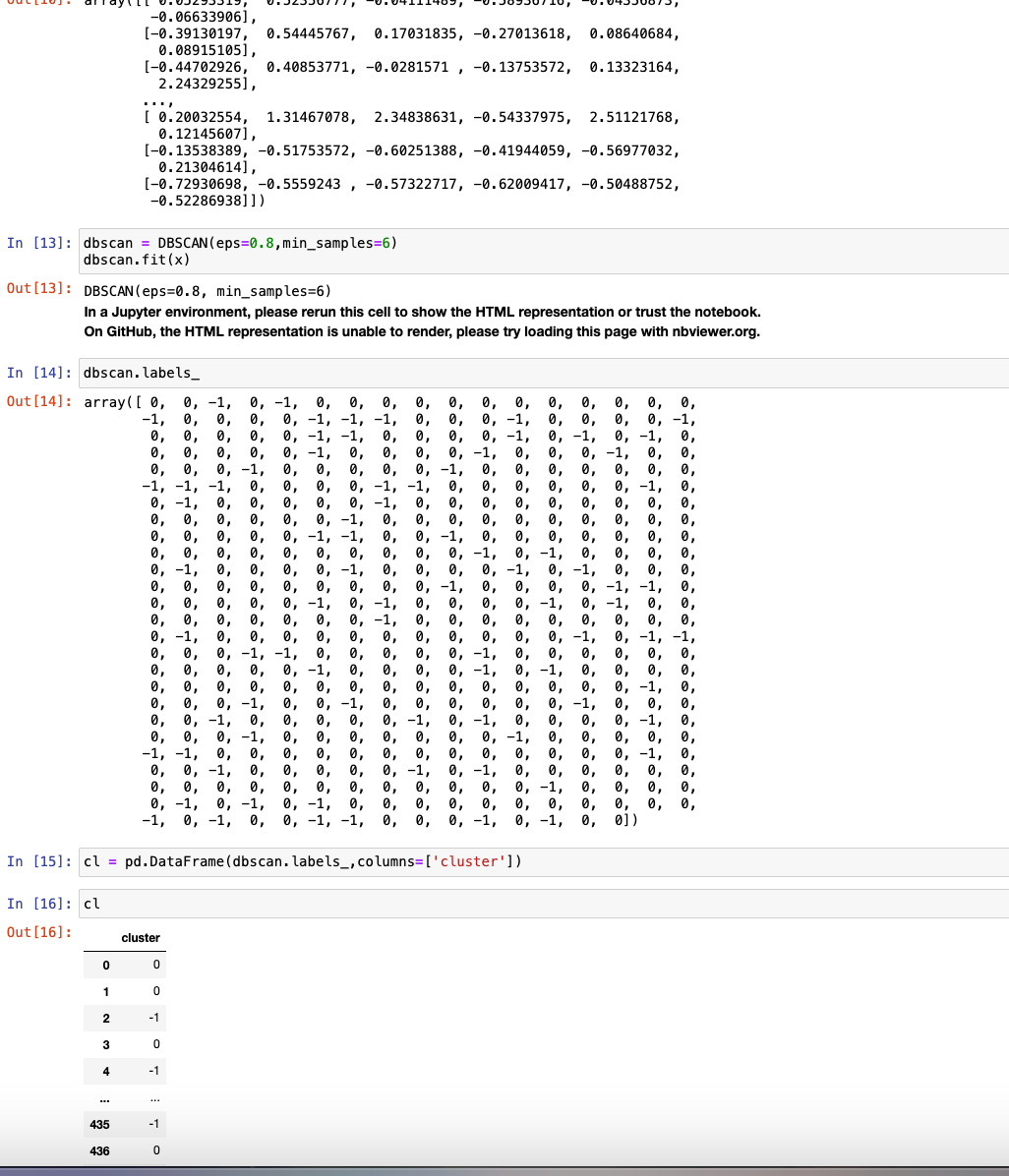
* DBSCAN, or Density-Based Spatial Clustering of Applications with Noise, is a clustering algorithm that identifies clusters of varying shapes and sizes in a dataset based on the density of data points.
* Unlike methods such as K-means, DBSCAN does not require specifying the number of clusters beforehand.
* Instead, it defines clusters as areas where there are many data points densely packed together, separated by areas of lower density (which may correspond to noise or outliers).
* In DBSCAN, two parameters are essential: epsilon (ε), which defines the radius within which to search for neighboring points, and minPts, the minimum number of points required within this radius to consider a point as a core point.
* Points that do not meet this criterion but are within the epsilon radius of a core point are considered border points, while points that do not meet either criterion are considered noise.

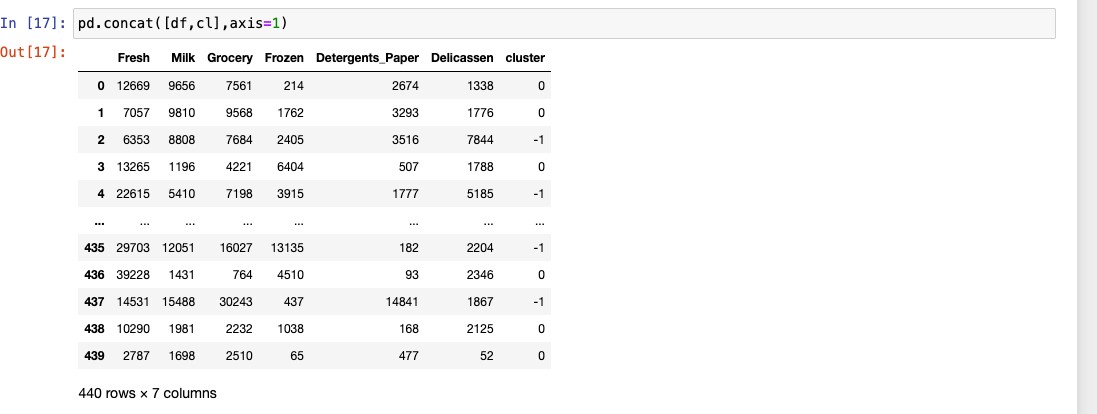
## Working:

* 1. Find all the neighbor points within eps and identify the core points or visited with more than MinPts neighbors.
  2. For each core point if it is not already assigned to a cluster, create a new cluster.
  3. Find recursively all its density-connected points and assign them to the same cluster as the core point.
  4. Iterate through the remaining unvisited points in the dataset. Those points that do not belong to any cluster are noise.

## Code:







**Conclusion:**

* In conclusion, the experiment validates DBSCAN's efficacy in clustering diverse datasets.
* Its parameter flexibility allows accurate clustering without prior cluster count knowledge.
* However, parameter tuning for optimal results may be challenging, necessitating domain expertise or further testing.
* Overall, DBSCAN proves invaluable for exploring datasets with varied structures and densities.

# PART B – ASSOCIATION

**Aim:** To study and analyse association.

**Software Used:** Jupyter Notebook.

## Theory:

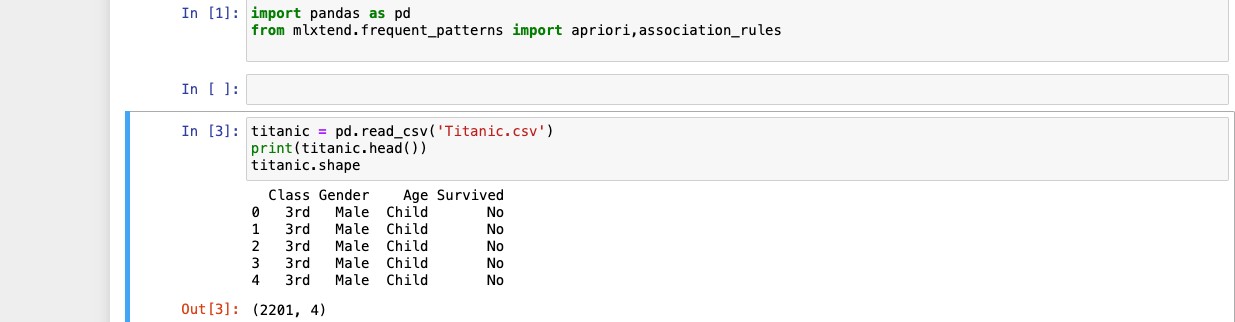
Association:

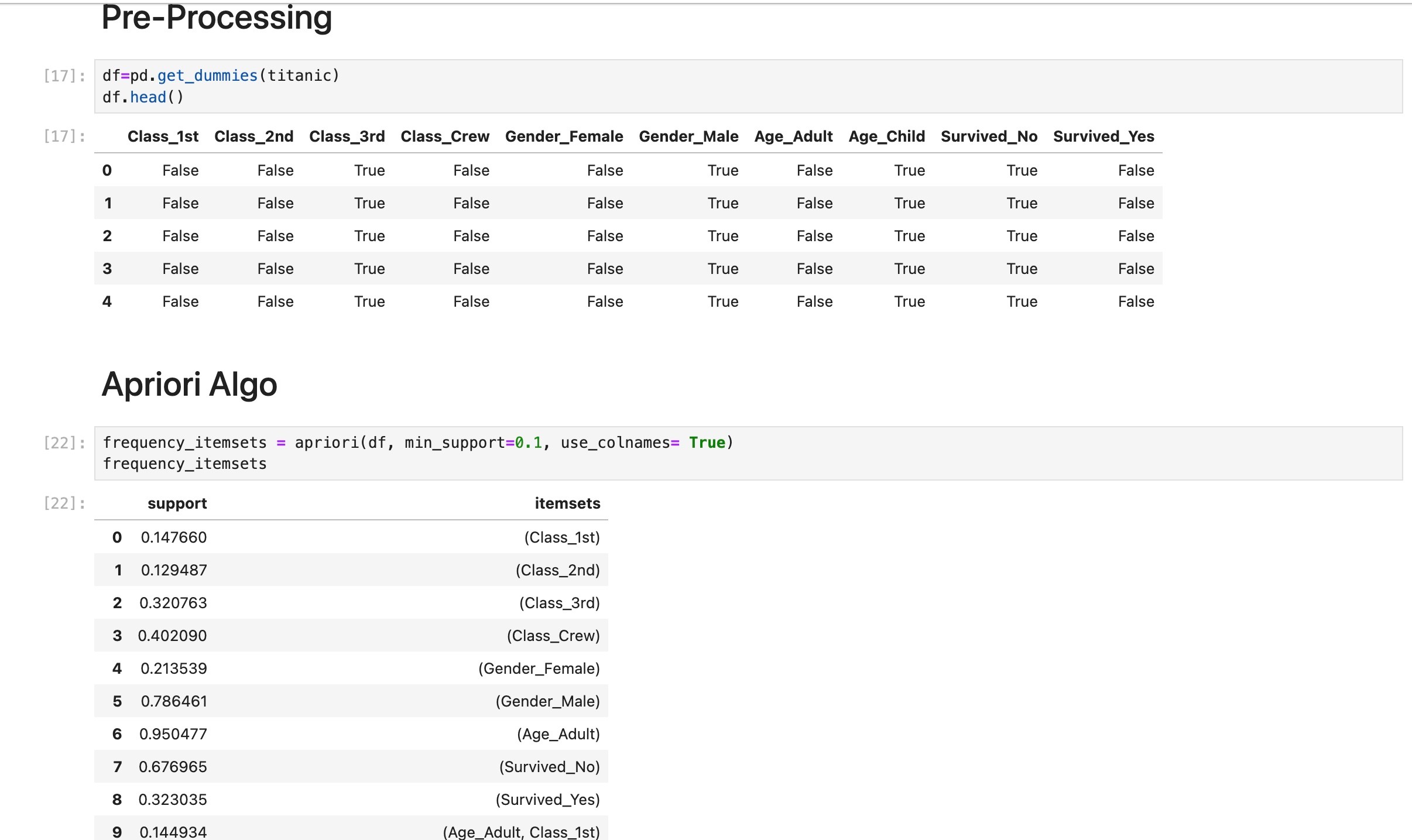
* Association in data science refers to identifying relationships or patterns between variables in a dataset.
* This process involves discovering associations, dependencies, or correlations between

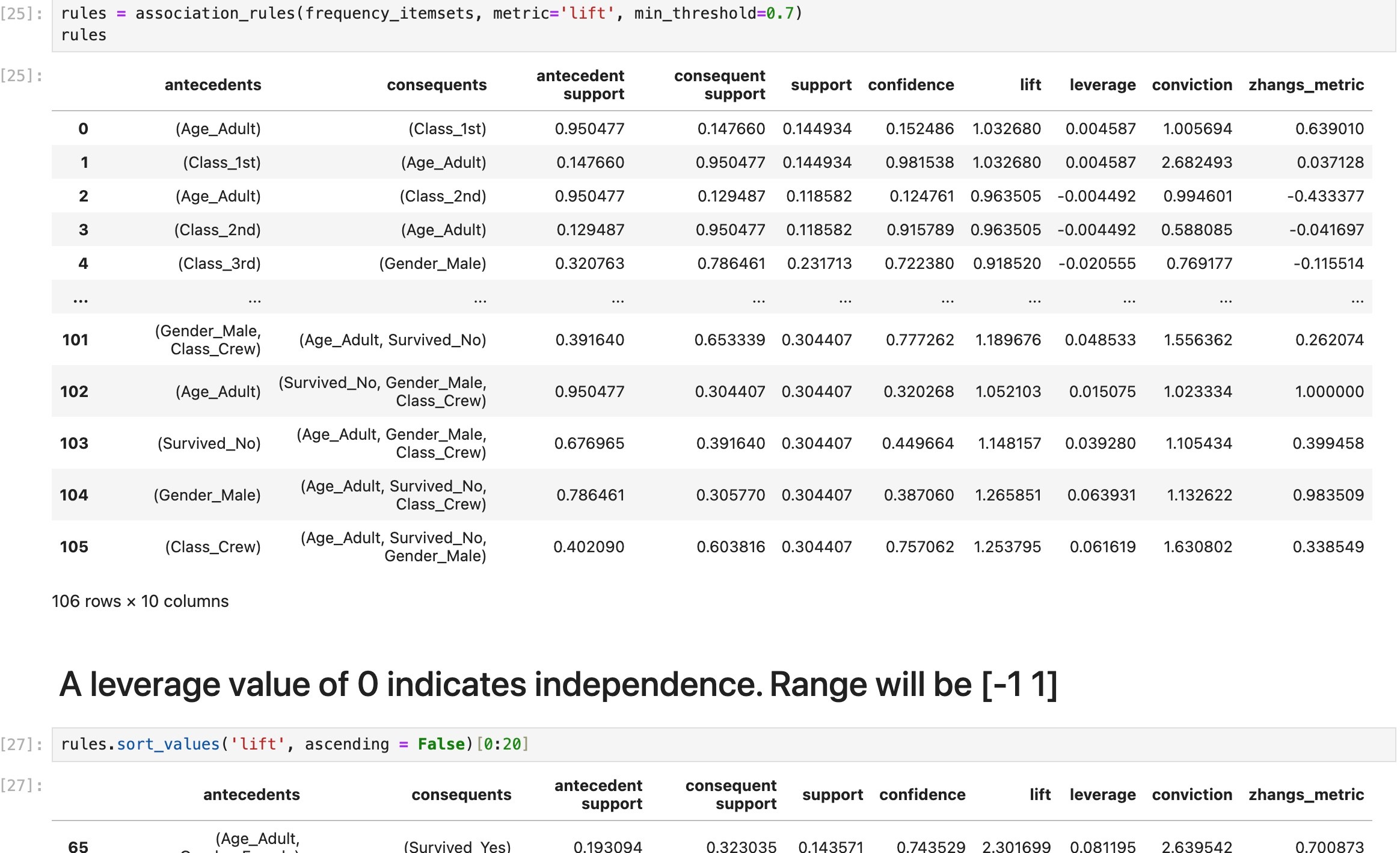
different attributes or items.

* Techniques such as association rule mining, commonly used for market basket analysis, aim to uncover frequent item sets or rules that describe relationships between items in transactional data.
* By identifying associations in data, businesses can gain valuable insights into consumer behavior, product affinities, and market trends.
* This knowledge can inform decision-making processes, such as product placement, targeted marketing campaigns, and inventory management strategies.

## Code:









**Conclusion:**

* In summary, association rule mining stands as a valuable tool for unveiling significant patterns and connections within extensive datasets, empowering businesses to enhance decision-making and refine strategies.
* Through the identification of frequent itemset and the formulation of association rules grounded in metrics like support, confidence, and lift, enterprises can extract actionable

insights into customer behavior, market dynamics, and other critical data facets.

* Nonetheless, meticulous attention to parameters and potential hurdles is essential to ascertain the practical relevance and trustworthiness of the unearthed associations in real-world scenarios.

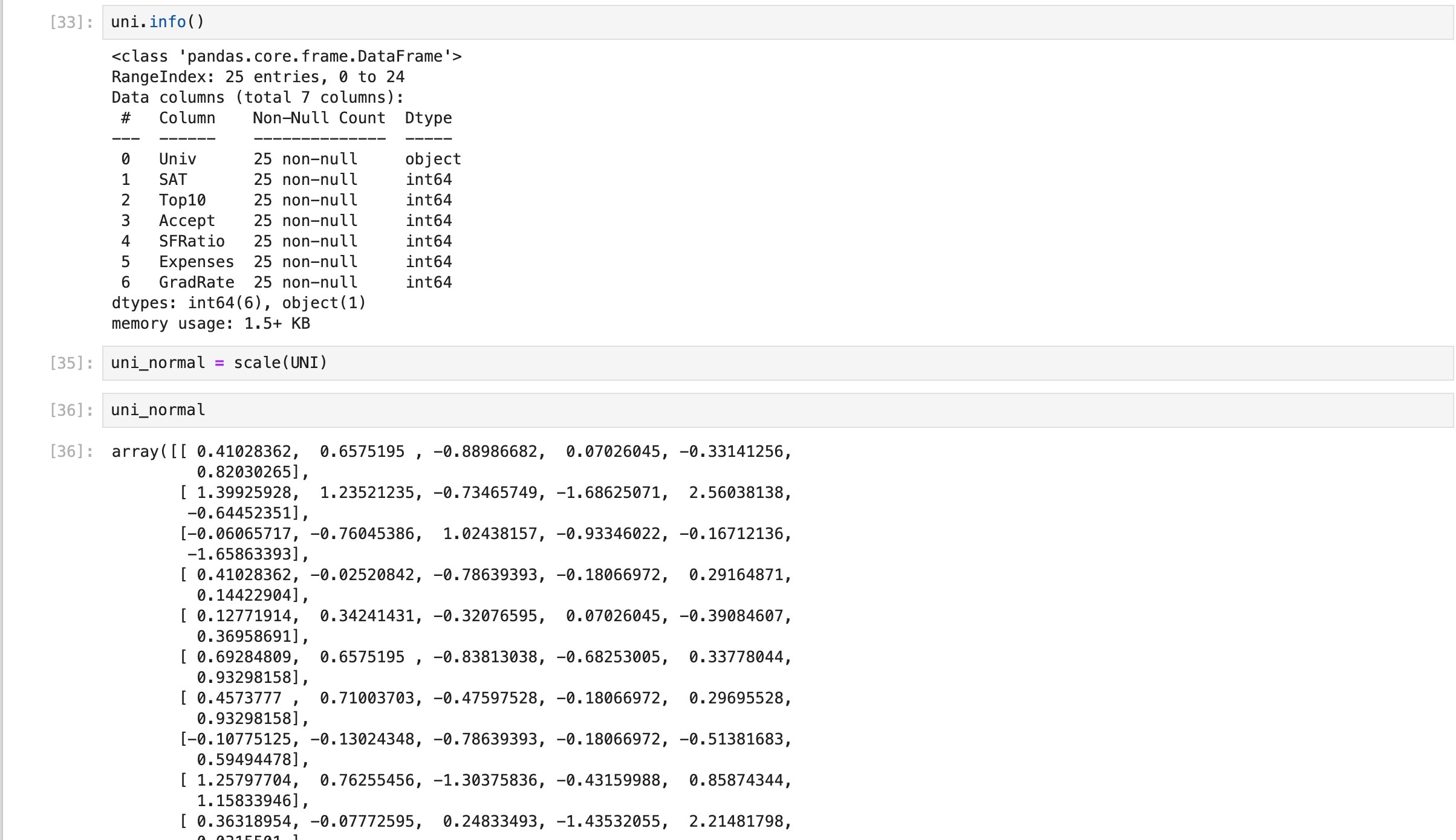
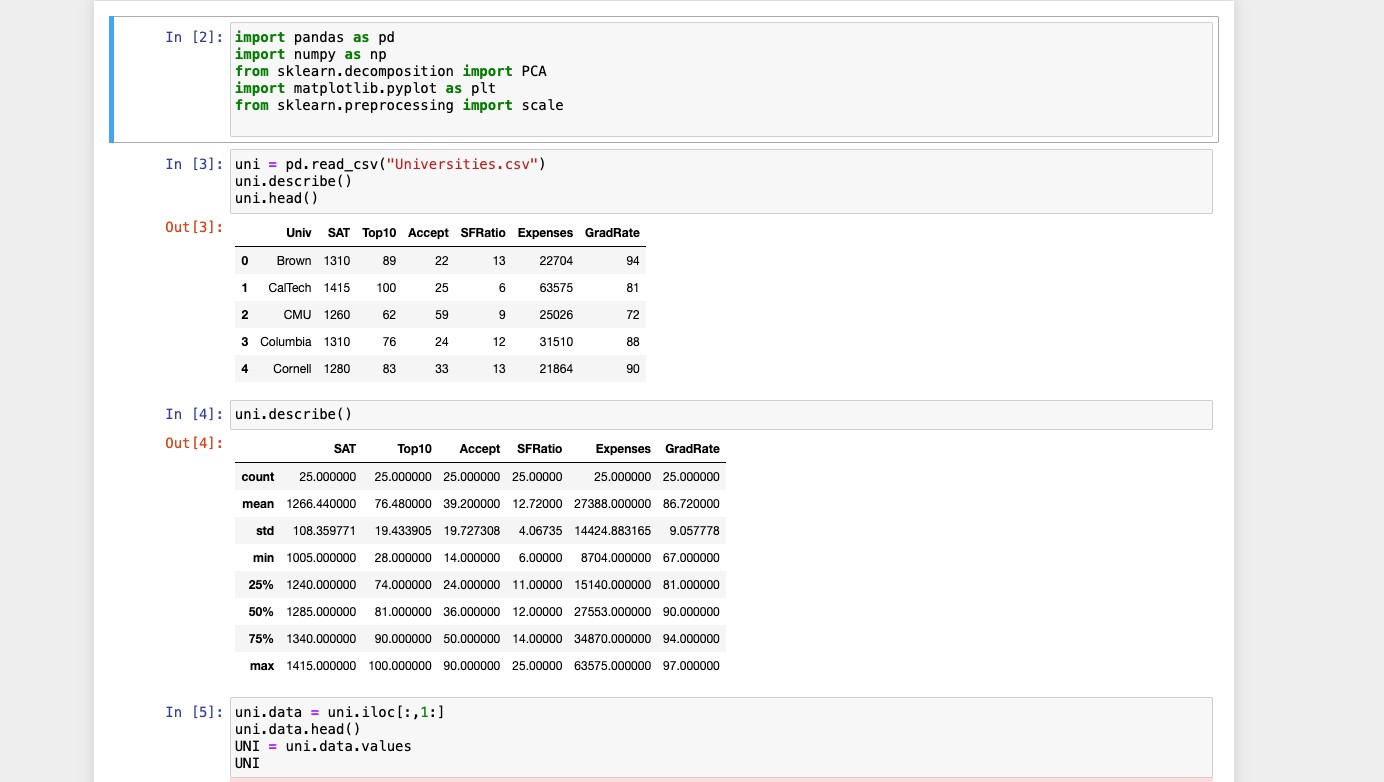
# PART C– PCA

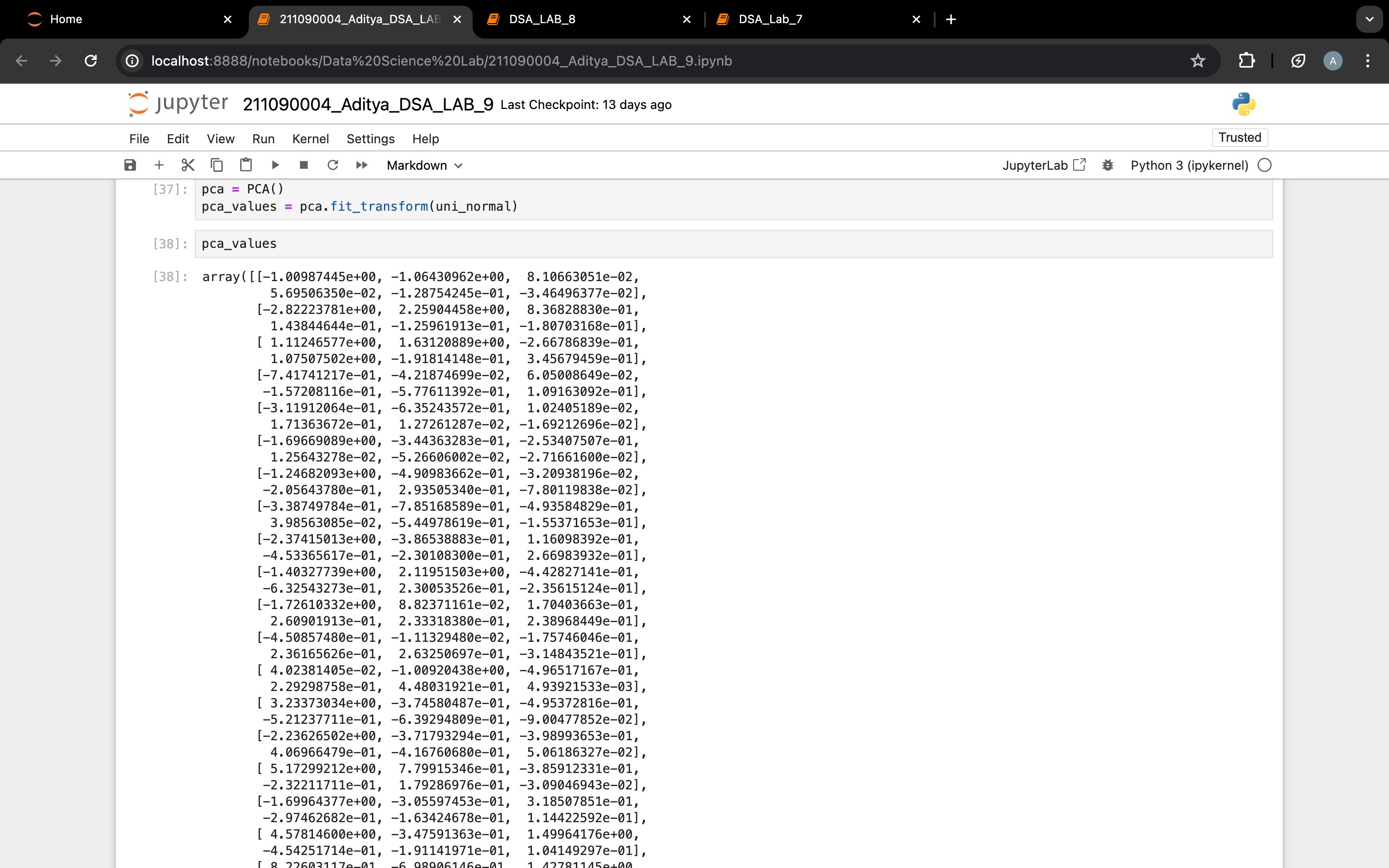
**Aim:** To study and analyse PCA. **Software Used:** Jupyter Notebook. **Theory:**

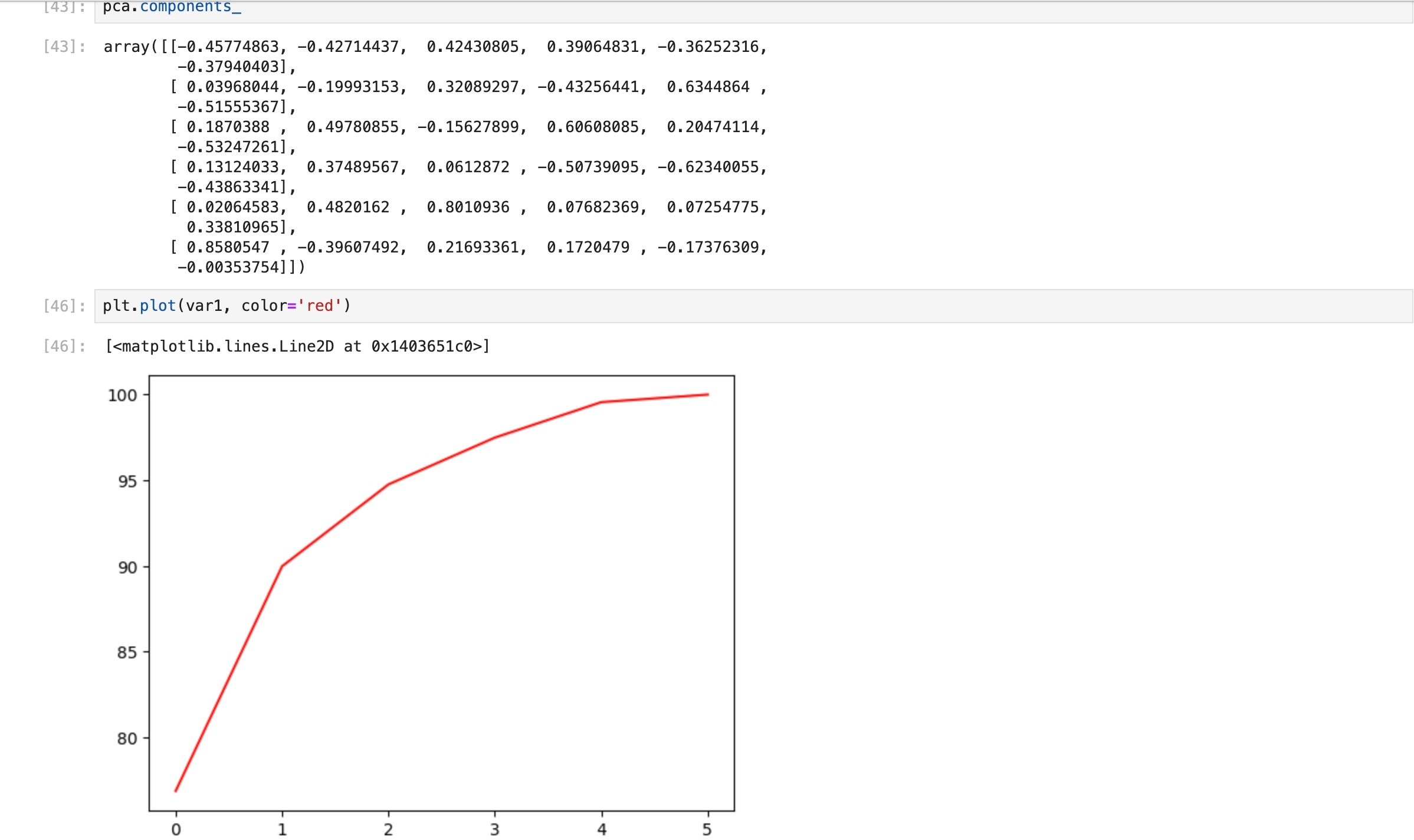
PCA(Principle Component Analysis):

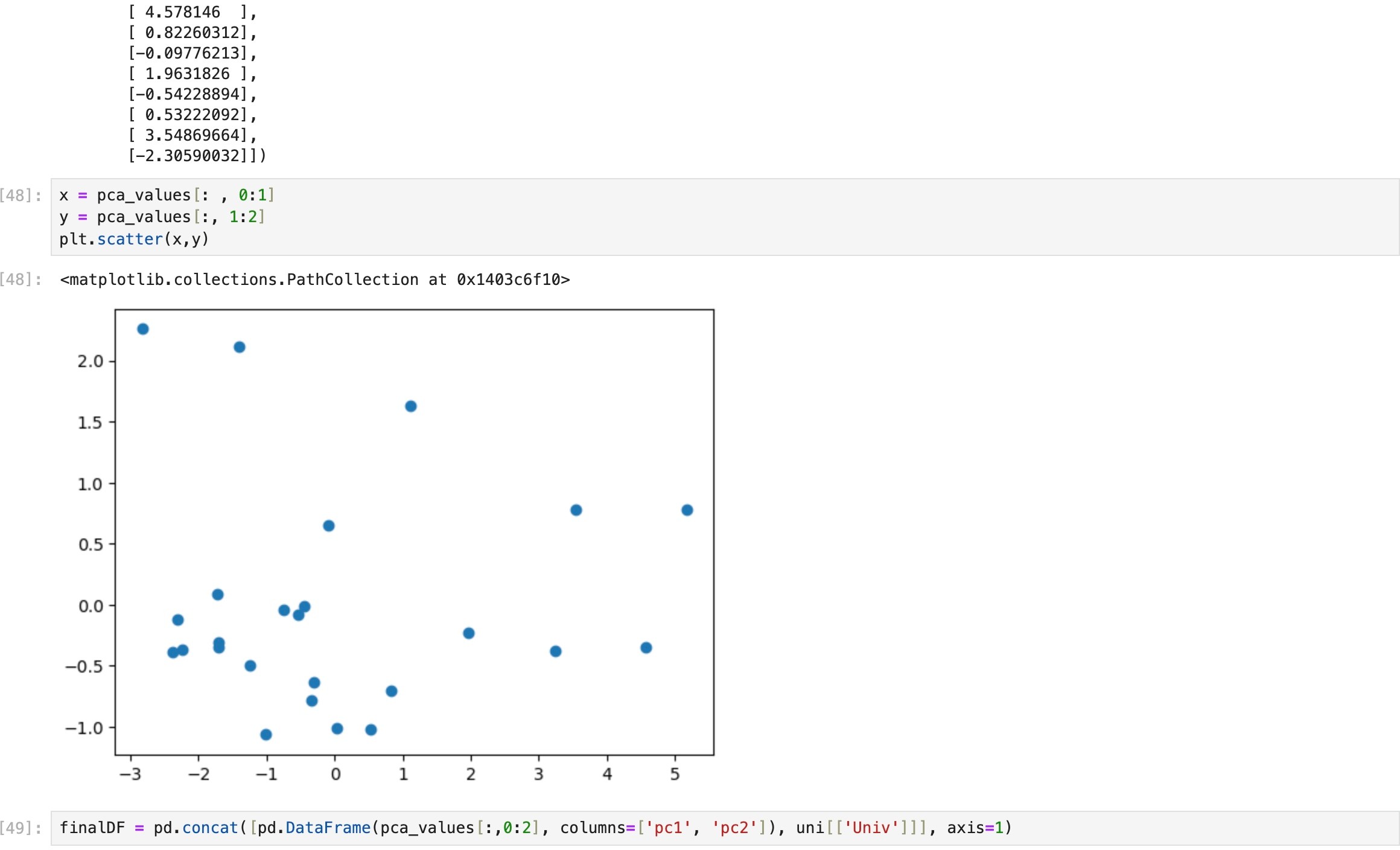
* As the number of features or dimensions in a dataset increases, the amount of data required to obtain a statistically significant result increases exponentially.
* This can lead to issues such as overfitting, increased computation time, and reduced accuracy of machine learning models this is known as the curse of dimensionality problems that arise while working with high-dimensional data.
* As the number of dimensions increases, the number of possible combinations of features increases exponentially, which makes it computationally difficult to obtain a representative sample of the data and it becomes expensive to perform tasks such as clustering or classification.
* So, PCA is used to reduce the number of input features while retaining most of the original information as much as possible.
* Principal Component Analysis (PCA) is a statistical procedure that uses an orthogonal transformation that converts a set of correlated variables to a set of uncorrelated variables.
* PCA is the most widely used tool in exploratory data analysis and in machine learning for predictive models.
* Moreover, it is an unsupervised learning algorithm technique used to examine the interrelations among a set of variables. It is also known as a general factor analysis where regression determines a line of best fit.
* The main goal is to reduce the dimensionality of a dataset while preserving the most important patterns or relationships between the variables without any prior knowledge of the target variables.

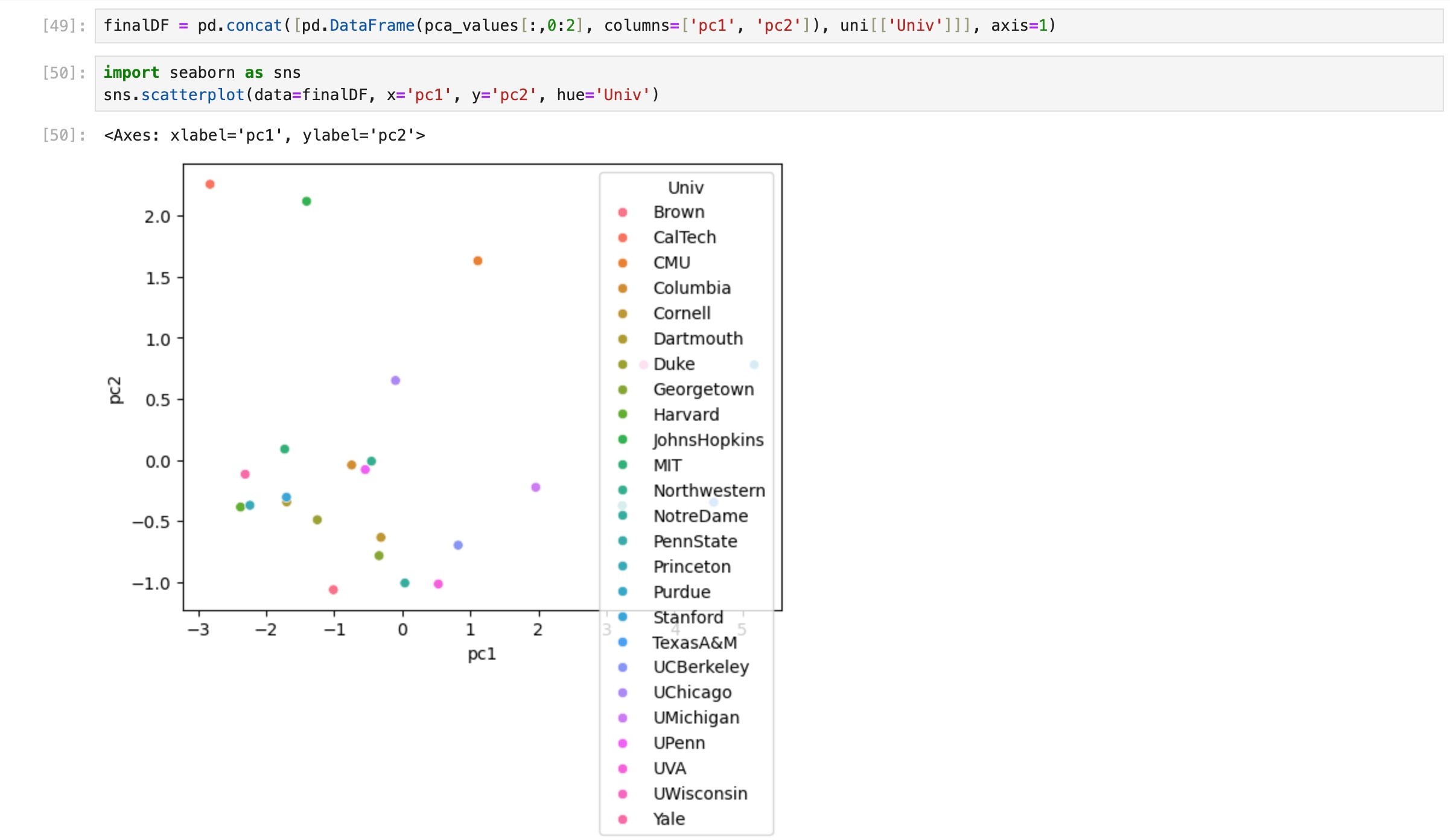
## Code:











**Conclusion:**

* In summary, Principal Component Analysis (PCA) provides a valuable technique for streamlining complex datasets through dimensionality reduction and feature extraction.
* This process retains maximal information while simplifying data, facilitating enhanced visualization, interpretation, and analysis of patterns.
* However, to effectively employ PCA in data analysis, one must acknowledge its assumptions and limitations, including interpretability loss and sensitivity to scaling.