applied sciences

MDPI

Article

Data Analysis for Information Discovery

Alberto Amato and Vincenzo Di Lecce *

Department of Electrical and Information Engineering, Politecnico di Bari, 70125 Bari, Italy
* Correspondence: vincenzo.dilecce@poliba.it

Abstract: Artificial intelligence applications are becoming increasingly popular and are producing better results in many areas of research. The quality of the results depends on the quantity of data and its information content. In recent years, the amount of data available has increased significantly, but this does not always mean more information and therefore better results. The aim of this work is to evaluate the effects of a new data preprocessing method for machine learning. This method was designed for sparce matrix approximation, and it is called semi-pivoted QR approximation (SPQR). To best of our knowledge, it has never been applied to data preprocessing in machine learning algorithms. This method works as a feature selection algorithm, and in this work, an evaluation of its effects on the performance of an unsupervised clustering algorithm is proposed. The obtained results are compared to those obtained using, as preprocessing algorithm, principal component analysis (PCA). These two methods have been applied to various publicly available datasets. The obtained results show that the SPQR algorithm can achieve results comparable to those obtained using PCA without introducing any transformation of the original dataset.

Keywords: data analysis; PCA; SPQR; FCM

check for updates

Citation: Amato, A.; Di Lecce, V. Data Analysis for Information Discovery. *Appl. Sci.* **2023**, *13*, 3481. https://doi.org/10.3390/app13063481

Academic Editor: Sławomir Nowaczyk

Received: 19 January 2023
Revised: 26 February 2023
Accepted: 6 March 2023
Published: 9 March 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Considering generic systems (natural or artificial phenomena, processes, etc.) and the relationships existing among inputs and outputs, they can be represented physically, mathematically, or logically. Anyhow, they create relationships among data (input and output).

In many phenomena, inputs to the system can be modelled as random variables. If our model is mathematical, this is a quantitative representation of a natural phenomenon. Similar to all other models used in science, its purpose is to represent, as incisively as possible, a given object, real phenomenon, or set of phenomena (mathematical model of a physical, chemical, or biological system).

Often, the model is an approximate representation of reality, but it is meaningful to conduce the analysis or prognosis.

Mathematical models are widely used in all areas of science. There are various mathematical tools in use from combinatorics to infinitesimal calculus. For example, for many phenomena, a very concise and intuitive description can be formulated immediately through differential equations.

For the sake of simplicity, we will speak of data input and data output from the model, regardless of the nature of the characteristic transfer function (e.g., linear dynamic systems, etc.).

In the present case, we consider the input and output data as discrete random variables, i.e.,

$$p(x) = P(X = x) \tag{1}$$

A random phenomenon (a phenomenon that is characterizable by a random variable) can be described in terms of the probability distribution and its parameters, such as the expected value and variance.

The relevance of variance in the study of models is widely used, for example, think of KLT (Karhunen–Loeve transform) [1], PCA, and its variants.

PCA is often used for dimensionality reduction. This process aims to reduce a large number of variables describing a dataset to a smaller number of latent variables, limiting the loss of information as much as possible [2]. In [3], authors used PCA in a computational chain to identify and remove noisy data from a dataset before addressing the classification problem of an artificial neural network (ANN). In [4] authors used PCA in a method to improve image classification performance by exploring the utilization of collective class characteristics to establish a statistically weighted algorithm and combined this weight with PCA to enhance the discrimination ability.

From this point of view, it is possible to say that giving input to a model original dataset and a dataset reduced using PCA, the model will produce almost the same results.

To test this hypothesis, in this work, a widely used algorithm (PCA) and an algorithm under our evaluation (semi-pivoted QR approximation SPQR [5,6]) have been used to reduce the dimensionality of the original data of various publicly available databases [7]. On the other hand, manipulating input data is one of the factors that can introduce uncertainty in numerical models. This is an interesting problem that is studied in the field of uncertainty analysis. Uncertainty analysis aims at quantifying the variability of the output that is due to the variability of the input. Some interesting approaches to this problem can be found in [8,9].

The SPQR method was presented in [5] and, to best of our knowledge, it has never been applied to dataset dimensionality reduction for machine learning algorithms.

Fuzzy clustering and silhouette analysis were used to compare the results. The performances obtained were compared among them, showing strong variations among the methods, highlighting the strong impact that preprocessing techniques can have on the performances of machine learning algorithms.

In further development of this, the information loss in the case of using the well-known PCA compared to the SPQR algorithm will be evaluated. This analysis was carried out on 10 public databases from the UCI site [7], and for each database, the following procedure was run:

- Fuzzy clustering on the raw database and performance evaluation with silhouette analysis;
- Fuzzy clustering on the preprocessed database with PCA and performance evaluation with silhouette analysis;
- Fuzzy clustering on the preprocessed database with SPQR and performance evaluation with silhouette analysis;
- All the previous tests performed on normalized data.

The remaining part of this paper is so organized: Section 2 reports a brief overview of related works, Section 3 describes the ten databases used for the tests, and Sections 4 and 5 describe, respectively, clustering and silhouette algorithms. The SPQR and PCA algorithms are described in Sections 6 and 7, respectively. Experiments and results are reported in Section 8, and conclusions and final remarks are in Section 9.

2. Related Works

Modern applications generate large amounts of data that are not always relevant. The transition between data and information is a topic of great interest to study. Many algorithms try to increase the density of knowledge from this data used in machine learning. We want to have less data in the original dataset, but with the same information or minimal information loss. To improve the computational efficiency of these algorithms, data preprocessing techniques are used to filter the data. Typically, statistics-based algorithms such as PCA are used.

Formally, PCA is a statistical technique for reducing the dimensionality of a dataset, linearly transforming the data into a new coordinate system, and preserving the maximum amount of information, often used to enable visualization of multidimensional data [10].

In our opinion, these techniques have a strong impact on the final performance of these algorithms, and therefore, we believe that push-button solutions should clear the way for more accurate analyses. The remaining part of this article will also focus on limitations and problems encountered in these applications.

Typically, PCA is used assuming that the relationships between variables are linear and comparable in terms of magnitude, i.e., they are scaled numerically. Several dimension reduction techniques have been introduced to handle nonlinear relationships, such as Isomap [11], locally linear embedding (LLE) [12], Hessian LLE [13], Laplacian eigenmaps [14], and its variants [15], including kernel PCA [16].

These methodologies discover the inherent geometric structure of high-dimensional data. In [17], authors showed that high-dimensional spaces are sparse and suffer from distance concentration.

This challenge makes the discovery of the intrinsic geometric structure nontrivial. In nonlinear dimension reduction (NDR), distance covariance is used for nonlinear relations, while the sparsity of high-dimensionality space is addressed by evaluating group dependencies.

To solve the problem of computational disadvantages and imperfect estimations in large-scale scenarios, divide-and-conquer mechanisms have been proposed for dimensionality reduction in [18–21]. In these works, the authors have shown that, in many cases, well-known application-specific relations and natural groupings can be exploited for efficient size reduction.

However, such approaches [18–21], require prior knowledge of these relationships to enable organization, which may not be available in big data scenarios. A divide-and-conquer approach is also presented in these studies.

Moreover, in traditional dimensionality reduction approaches [11–24], the basic principle is to perform one-step mapping from an upper-dimensional space to a lower-dimensional space. There is no mechanism for incorporating new information into the analysis structure when new data are available. In these scenarios, it would be necessary to recalculate the dimension reduction [22–24], which is inefficient.

Although parameter aggregation methods facilitate the Pearson correlation and/or covariance updating [23] with the corresponding single-value decomposition (SVD) [25], they are focused on computational efficiency when all data are available.

However, unlike in [18–21], NDR allows for SVD results to be used to perform group organization. Moreover, if information on such relationships is available, the generic organizational structure proposed here allows for this information to be incorporated at the first step.

First, such an approach is computationally poor when the amount of original data is very large. It is also susceptible to improper estimation due to the presence of large amounts of noise and redundancies [26].

Dimensionality reduction approaches require the practitioner to specify the number of dimensions to be extracted from the data. In some algorithms, the user simply specifies the percentage of information to be stored in the NDR, and the algorithm estimates the number of dimensions to be extracted from the data. Unfortunately, this is not the case in big data scenarios; these methods can be computationally inefficient.

The algorithm studied and proposed in this paper, called “semi-pivoted QR approximation” (SPQR), is an efficient deterministic method to reduce a given matrix to its most important columns by estimating its significance and specific information content. It was introduced by Stewart [5,27].

As the name highlights, the approach is based on the QR decomposition that expresses a matrix A as the product of an orthogonal matrix Q and an upper triangular matrix R. These factors are obtained by the Gram–Schmidt algorithm to orthonormalize the columns of A, one at a time, from first to last. This procedure is preferred because it uses Pivoted QR: it differs in that the Gram–Schmidt procedure takes the largest remaining column at

the beginning of each new step [6]. The new step is taken, and then a permutation matrix P is operated such that:

$$A - P = Q - R \quad (2)$$

3. Database Description

All the experiments carried out in this work have used public databases downloaded from [7]. This repository contains a large number of datasets. In this work, all the experiments have been carried out on ten datasets. These datasets are very different among them in many aspects: number of features, number of instances, semantic content, etc. This choice improves the generalizability of the obtained results.

In particular, the used databases are:

1. Gender Gap in Spanish WP Dataset [28]: Dataset used to estimate the number of women editors and their editing practices in the Spanish Wikipedia. It is composed of 21 attributes and 4746 instances;
2. TUANDROMD (Tezpur University Android Malware Dataset) Dataset [29]: This dataset contains 4465 instances and 241 attributes. The target attribute for classification is a category (malware vs. goodware);
3. Room Occupancy Estimation Dataset [30]: this dataset contains 10,129 instances and 16 attributes, and it is used to estimate the occupation level of a room. The setup consisted of seven sensor nodes and one edge node in a star configuration with the sensor nodes transmitting data to the edge every 30 s using wireless transceivers. Each sensor node contained various sensors such as temperature, light, sound, CO_2, and digital passive infrared (PIR);
4. Myocardial Infarction Complications Dataset [31]: This dataset contains 1700 instances and 124 attributes. The main application of this database is to predict complications of myocardial infarction based on information about the patient at the time of admission and on the third day of the hospital period;
5. Wine Dataset [7]: This dataset is used to recognize a specific wine using the concentration of 13 chemical parameters. It is composed of 13 features and 178 instances;
6. Dry Bean Dataset [32]: This dataset contains 16 visual features extracted from 13,611 images of grains of 7 different dry beans. The used visual features are in 12 dimensions and 4 shape forms;
7. APS Failure at Scania Trucks Dataset [7]: This dataset contains 60,000 instances and 171 features. The dataset consists of data collected from heavy Scania trucks in everyday usage. The system in focus is the air pressure system (APS), which generates pressurized air that is utilized in various functions in the truck, such as braking and gear changes;
8. Multiple Features Dataset [7]: This dataset consists of the features of handwritten numerals ('0'–'9') extracted from a collection of Dutch utility maps. It is composed of 649 features and 2000 instances;
9. Relative Location of CT Slices on Axial Axis Dataset [7]: This dataset consists of 384 features extracted from 53,500 CT images. The class variable is numeric and denotes the relative location of the CT slice on the axial axis of the human body;
10. Mice Protein Expression Dataset [33]: This dataset contains 1080 instances and 82 features. The dataset consists of the expression levels of 77 proteins/protein modifications that produced detectable signals in the nuclear fraction of the cortex.

In all the experiments, for each database, only the numerical features have been considered.

4. Clustering

The term "clustering" refers to a wide family of learning algorithms that comprise unsupervised, supervised [34], and semi-supervised [35] learning techniques. These algorithms implement different approaches (hierarchical, partitional, grid, density, and model

based [34]) to solve the same classification problem, namely, grouping objects according to a given similarity criterion.

In this work, the authors used the fuzzy C-means (FCM) algorithm [36,37]. This clustering algorithm generates fuzzy partitions and prototypes for any set of numerical data. It works by optimizing a generalized least squares objective function:

$$Q = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^{m} \parallel x_k - v_i \parallel^2 \quad (3)$$

where

- $||.||$ is a distance function such as: Euclidean, Mahalanobis, etc.;
- $v_1, v_2, \ldots, v_c$ are the centroids of the clusters, also called prototypes;
- $\mathbf{X} = \{x_1, x_2, \ldots, x_N\}$ is the set of points to be clustered;
- $\mathbf{U} = [u_{ik}]$ is the partition matrix;
- c is the number of clusters;
- N is the number of points to be clustered;
- i is an index that varies from 1 to c;
- k is an index that varies from 1 to N;
- "m" is a coefficient called the "fuzzification coefficient". It is greater than 1, and it is responsible for the level of "fuzziness" of the partition matrix. In other words, it controls the level of fuzziness with which each point belongs to the various clusters.

This algorithm allows for the discovery of the inner structure of a given dataset **X**, minimizing the objective function **Q** according to the given prototypes and the partition matrix **U**. The minimization process is iterative. It works by updating the values of the partition matrix and the prototypes until a given criterion of stop is reached.

For example, considering two partition matrices, U and U′, obtained in two consecutive iterations, the procedure may stop when the quantity:

$$||U - U'|| = \max_{i,k} |u_{ik} - u'_{ik}| \quad (4)$$

gets smaller than some predefined positive threshold ε.

It should be noticed that the nature of the optimization function leads to a solution that somehow reflects the geometry of the original dataset.

On the other hand, the hidden structure of the dataset, discovered by the clustering algorithm, can be used to classify other elements added to the set after the initial clustering. This task can be accomplished applying the following rules:

- The "anchor points" of the classifier are the prototypes of the clusters;
- Each cluster defines a class;
- A point x belongs to a class defined by the cluster with prototype v_j if:

$$j = \arg\left(min_i ||x - v_i||^2\right) \quad (5)$$

5. Silhouette Method for Clustering Evaluation

The performance evaluation of a clustering algorithm is not a trivial task. Indeed, when using a clustering algorithm, it is possible to be in one of the following situations:

- The correct solution is known: in this case, the classification of each point is known a priori, and the classification performance of the clustering algorithm can be computed counting the number of misclassified patterns (error rate);
- The correct solution is subjective: in this case, there is no true ground against which to evaluate the results of the clustering algorithm. Since the classification is subjective, there is no universally acceptable solution, so the classification task falls into the problem of the semantic gap [38];
- The correct solution is unknown: also in this case, there is no true ground against which to evaluate the results of the clustering algorithm. To face this problem, there

are various approaches as reported in [39]. On the other hand, in the latest years, the silhouette parameter has become a more commonly used method for assessing the quality of clusters [40].

In this work the silhouette parameter has been used to evaluate the clustering performance. This method is based on a comparative evaluation of the similarity level of each object in relation to its cluster (tightness) and to the other clusters (separation). This parameter is defined as follows: if y is a point belonging to cluster A, then $t(y)$ is the mean distance between y and all other points of A. Let us now consider any cluster B, different from A, and compute the average distance between y and all points of B($d(y, B)$). Once we have computed $d(y, B)$ for each cluster B such that B $\neq$ A, we select the smallest of these numbers and denote it by:

$$v(y) = \min_{A \neq B} d(y, B) \tag{6}$$

Starting from these considerations, the silhouette for the point y is defined as shown in the following formula:

$$s(y) = \begin{cases} 1 - \frac{t(y)}{v(y)} & if\ t(y) < v(y) \\ 0 & if\ t(y) = v(y) \\ \frac{v(y)}{t(y)} - 1 & if\ t(y) > v(y) \end{cases} \tag{7}$$

From this definition, it is possible to say that for each point x into the dataset:

$$-1 < s(y) < +1$$

An in-depth analysis of this parameter is reported in [40].

On the other hand, using this method makes it possible to show the obtained results using a graphical representation highlighting, for each point, how well it has been classified. Furthermore, the flexibility of this method should be highlighted due to the fact that it is possible to use any kind of distance metric, such as Mahalanobis, Euclidean, Manattan, etc.

All these features give to the silhouette parameters a strong appeal when it is required to evaluate the performance of a clustering algorithm applied to datasets for which there is no a-priori knowledge (this is also the case study of this paper).

Another interesting application of the silhouette method is as a reference guide in discovering the optimal number of clusters for a given dataset. This task can be carried out by implementing an iterative process composed of the following steps:

1. Run the clustering algorithm using a certain number of clusters "C";
2. Compute the silhouette for the obtained clusters.

If there is a satisfying number of points with a good level of silhouette, then "C" can be considered a good number of clusters. Otherwise, change the value of "C" and return to Step 1.

6. SPQR Algorithm

Many data applications deal with the necessity to represent m objects characterized by n features. One of the most often used representations of this kind of data is a matrix **A** composed of m rows and n columns. In modern applications of data analysis, such as image analysis and environmental datasets, these matrices often have high dimensionalities. This fact leads to a series of difficulties in the processes of data mining, representation, communication, and storage.

In the latest years, many works in the field of feature selection [41] have demonstrated that, through analyzing a dataset, it is possible to identify and eliminate some redundant and/or irrelevant features. By applying feature selection methods, it is possible to achieve a series of advantages in data analysis processes, such as a decrease in the amount of data, an improvement in the prediction accuracy, pulling out important features, understanding the

attributes or variables easily, and finally reducing the execution time [41]. An interesting overview on feature section methods can be found in [41].

Many of these methods aim to approximate the matrix A by means of a "smaller" matrix obtained by combining its columns and rows. The drawback of these methods is that they usually yield dense factorizations, and more seriously, these terms are often much harder to interpret than the original ones. For example, truncating the SVD at k terms is one of the most common ways to obtain the "best" rank-k approximation of A when measured with respect to any unitarily invariant matrix norm. The drawback of this method is that it produces a representation of the dataset that is difficult to relate to the original dataset and to the processes that generated it. The same drawbacks are present in another widely used method for feature selection: principal component analysis (PCA).

Starting from these considerations, many methods to solve the column subset selection problem have been developed [42]. These methods try to find a subset of k actual columns of A, with k less than n, which "captures" most of the information of A with respect to the spectral or the Frobenius norm.

Essentially two classes of methods may be defined:

1. Randomized methods: these methods select the most representative columns in a matrix using probability distributions;
2. Deterministic methods: these methods select the columns in a deterministic way.

An effective deterministic method to reduce the matrix A to its more important columns is "semi-pivoted QR approximation" (SPQR) by Stewart [5,27,43]. As its name suggests, the key approach is the QR decomposition of A into an orthogonal matrix Q and a triangular matrix R. These factors are computed by the Gram–Schmidt algorithm for orthonormalizing the columns of A, one at a time, from first to last. In many situations *pivoted* QR is preferred: in practice, columns are exchanged at the start of each new stage to take the largest remaining column. In this way a permutation matrix P is built such that:

$$A \cdot P = Q \cdot R \tag{8}$$

When A is rank deficient, the column pivoting A·P is applied to improve the numerical accuracy. Moreover, the choice of P usually guarantees that R does not have increasing diagonal entries, a specific feature which is useful in the following. More in detail, we may partition the expression above as:

$$[\, B_1 \; B_2] = [\, Q_1 \; Q_2] \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix} \tag{9}$$

The following properties hold:

1. $B_1 = Q_1 \cdot R_{11}$
2. $||B_2 - Q_1 \cdot R_{12}|| = ||R_{22}||$

The semi-QR algorithm exploits these results to use the approximation:

$$A \cdot P \approx Q_1 \cdot [R_{11} \quad R_{12}] \tag{10}$$

that, thanks to Property 1 above, reproduces the first k columns of A·P exactly by introducing a quantifiable error (Property 2). An additional strength of this method is that the explicit computation of the nonsparse orthogonal matrix Q_1 is not required.

In practice, given a rank parameter k, the SPQR algorithm gives k columns of A whose span approximates the column space of A; they form the matrix B_1 of dimension $m \times k$, while the factor R_{11} contains the coefficients of the column orthogonalization.

7. PCA

Principal component analysis (PCA) is a well-known method often used to reduce the dimensionality of large databases. Reducing the dimensions of a dataset introduces a loss in accuracy, but on the other hand, it improves the efficiency of algorithms of data analysis,

such as data exploration, data visualization, machine learning, etc. From this perspective, when using PCA, it is required to find a trade-off between performance improvement and accuracy loss.

In [22], there is an in-depth analysis of the PCA method, while in this section, a brief operative description is proposed.

Essentially, PCA can be seen as a five-step process:

1. Standardization: In this step, the range of each initial variable (each column of the dataset is a variable) is standardized so that each variable contributes equally to the overall analysis;
2. Covariance matrix computation: The goal of this step is to compute the degree of the relationship among the variables of the dataset;
3. Identification of the principal components: These are new variables obtained as a linear combination or mixture of the initial variables. These new variables have the following properties: they are uncorrelated and most of the information contained in the initial variables is compressed in the first components. This allows for the reduction in dimensionality without losing too much information, discarding the components with low information and considering the remaining components as the new variables. To find these principal components, eigenvectors and eigenvalues of the covariance matrix are computed because they are the directions of the axes where there is the most variance (most information);
4. Feature vector selection: In this step, the eigenvectors are disposed in descending order by their eigenvalues, allowing it to find the principal components in order of significance. In this way, it is possible to choose the number of components to keep for further analysis, discarding those of lesser significance (of low eigenvalues);
5. Recast the data along the principal component axes: Once the principal components have been chosen, it is necessary to recast on them the original dataset.

8. Experiments and Results

In this section a brief description of the carried-out experiments is reported. For each database described in Section 3, a clustering analysis was conducted using the FCM algorithm, varying the number of clusters from 10 to 50. The results of each clustering were evaluated using the silhouette parameter. For each database, the percentages of points with a silhouette level greater than 0.7, 0.8, and 0.9 are reported. These analyses were carried out six times on each database using:

1. The raw data (namely, the original data stored in each database);
2. The dataset reduced using the PCA algorithm, losing less than 2% of the total variation in the dataset;
3. The dataset reduced using the SPQR algorithm, setting the same number of features used with PCA;
4. The raw data normalized between 0 and 1;
5. The normalized dataset reduced using the PCA algorithm, losing less than 2% of the total variation in the dataset;
6. The normalized dataset reduced using the SPQR algorithm, setting the same number of features used with PCA;
7. In the following, the obtained results for each database are reported.

8.1. Gender Gap in Spanish WP Dataset

8.1.1. Clustering and Silhouettes Using Raw Data

The original dataset is composed of 20 numerical features. Applying PCA to this dataset, losing less than 2% of the total variation in the data, produced a dataset with a single dimension.

The following Tables 1–3 show the obtained results in terms of silhouettes varying the number of clusters from 10 to 50 (see columns). Each row represents the percentage of points with a silhouette greater than a given threshold, that is 0.7, 0.8, and 0.9.

Table 1. Clustering performance in terms of silhouettes using original data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	43.95	37.44	43.28	50.46	60.81
0.8	21.60	16.29	29.31	39.40	51.45
0.9	0	0.40	16.75	35.92	37.340

Table 2. Clustering performance in terms of silhouettes using the reduced dataset with PCA.

Silhouette\N. Clusters	10	20	30	40	50
0.7	43.95	37.44	43.28	50.46	60.81
0.8	21.60	16.29	29.31	39.40	51.45
0.9	0	0.40	16.75	35.92	37.340

Table 3. Clustering performance in terms of silhouettes using the reduced dataset with SPQR.

Silhouette\N. Clusters	10	20	30	40	50
0.7	92.54	76.06	78.93	82.49	87.80
0.8	89.55	69.64	69.59	75.41	74.40
0.9	78.59	47.83	57.753	46.103	48.463

8.1.2. Clustering and Silhouettes Using Normalized Data

The original dataset was normalized between 0 and 1. Applying PCA to this dataset, losing less than 2% of the total variation in the data, produced a dataset with six dimensions.

The following Tables 4–6 show the obtained results in terms of silhouettes varying the number of clusters from 10 to 50 (see columns). Each row represents the percentage of points with a silhouette greater than a given threshold, that is 0.7, 0.8, and 0.9.

Table 4. Clustering performance in terms of silhouettes using normalized data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	0.06	0.08	0.17	0.08	0.23
0.8	0.06	0.08	0.10	0.08	0.15
0.9	0.06	0.08	0.06	0.06	0.15

Table 5. Clustering performance in terms of silhouettes using the reduced dataset with PCA applied to normalized data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	0	1.92	1.96	0.02	0.17
0.8	0	0.04	0	0.02	0.17
0.9	0	0.04	0	0.02	0.17

Table 6. Clustering performance in terms of silhouettes using the reduced dataset with SPQR applied to normalized data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	8.81	3.37	2.61	2.40	1.85
0.8	0	1.60	0	2.21	1.01
0.9	0	0	0	0.46	0.63

In these experiments, the results obtained using normalized data seem to be worse than those obtained with the raw data.

8.2. Room Occupancy Estimation Dataset

8.2.1. Clustering and Silhouettes Using Raw Data

The original dataset is composed of 20 numerical features. Applying PCA to this dataset, losing less than 2% of the total variation in the data, produced a dataset with two dimensions.

The following Tables 7–9 show the obtained results in terms of silhouettes varying the number of clusters from 10 to 50 (see columns). Each row represents the percentage of points with a silhouette greater than a given threshold, that is 0.7, 0.8, and 0.9.

Table 7. Clustering performance in terms of silhouettes using original data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	61.85	68.70	59.29	67.41	56.31
0.8	55.32	49.05	31.20	39.98	46.01
0.9	50.29	34.87	5.56	9.22	22.68

Table 8. Clustering performance in terms of silhouettes using the reduced dataset with PCA.

Silhouette\N. Clusters	10	20	30	40	50
0.7	67.64	68.19	61.77	61.23	56.06
0.8	58.49	47.50	43.93	44.03	45.42
0.9	48.98	34.78	7.918	6.74	16.00

Table 9. Clustering performance in terms of silhouettes using the reduced dataset with SPQR.

Silhouette\N. Clusters	10	20	30	40	50
0.7	63.34	54.01	62.71	62.11	57.87
0.8	46.83	35.16	46.13	26.06	40.93
0.9	4.68	13.31	2.59	3.29	19.58

8.2.2. Clustering and Silhouettes Using Normalized Data

The original dataset was normalized between 0 and 1. Applying PCA to this dataset, losing less than 2% of the total variation in the data, produced a dataset with five dimensions.

The following Tables 10–12 show the obtained results in terms of silhouettes varying the number of clusters from 10 to 50 (see columns). Each row represents the percentage of points with a silhouette greater than a given threshold, that is 0.7, 0.8, and 0.9.

In these experiments, the results obtained using normalized data seem to be slightly worse than those obtained with the raw data.

Table 10. Clustering performance in terms of silhouettes using normalized data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	15.03	20.39	13.74	19.47	23.40
0.8	6.17	5.67	4.04	8.10	8.65
0.9	0	0.01	0.02	1.03	0.03

Table 11. Clustering performance in terms of silhouettes using the reduced dataset with PCA applied to normalized data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	51.43	41.16	38.56	37.08	32.20
0.8	32.08	21.46	21.27	17.01	14.96
0.9	4.34	0.01	1.58	0.83	3.14

Table 12. Clustering performance in terms of silhouettes using the reduced dataset with SPQR applied to normalized data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	53.34	56.47	55.40	61.69	62.98
0.8	37.88	35.16	34.88	40.51	40.34
0.9	13.51	19.07	10.68	9.81	17.37

8.3. Myocardial Infarction Complications Dataset

8.3.1. Clustering and Silhouettes Using Raw Data

This dataset could be considered a sort of spare matrix. There are many 0s, and some features contain not-a-number values (NaN). The proposed results were obtained, selecting only the features without NaN values, obtaining a dataset composed of fourteen dimensions. Applying PCA to this dataset, losing less than 2% of the total variation in the data, produces a dataset with one dimension.

The following Tables 13–15 show the obtained results in terms of silhouettes varying the number of clusters from 10 to 50 (see columns). Each row represents the percentage of points with a silhouette greater than a given threshold, that is 0.7, 0.8, and 0.9.

Table 13. Clustering performance in terms of silhouettes using original data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	48.91	25.48	21.60	16.59	15.48
0.8	27.31	14.24	8.83	8.06	7.062
0.9	0	5.59	3.35	2.24	0.82

Table 14. Clustering performance in terms of silhouettes using the reduced dataset with PCA.

Silhouette\N. Clusters	10	20	30	40	50
0.7	62.45	64.04	66.27	64.92	67.10
0.8	52.44	50.85	52.62	51.68	54.21
0.9	23.19	24.48	25.37	23.90	27.78

Table 15. Clustering performance in terms of silhouettes using the reduced dataset with SPQR.

Silhouette\N. Clusters	10	20	30	40	50
0.7	63.39	61.86	59.98	59.86	59.74
0.8	50.68	47.38	46.44	46.08	47.14
0.9	22.31	19.36	18.07	17.83	17.83

8.3.2. Clustering and Silhouettes Using Normalized Data

The original dataset (without the features containing NaN values) was normalized between 0 and 1. Applying PCA to this dataset, losing less than 2% of the total variation in the data, produced a dataset with nine dimensions.

The following Tables 16–18 show the obtained results in terms of silhouettes varying the number of clusters from 10 to 50 (see columns). Each row represents the percentage of points with a silhouette greater than a given threshold, that is 0.7, 0.8, and 0.9.

Table 16. Clustering performance in terms of silhouettes using normalized data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	27.49	11.42	0.12	2.77	3.12
0.8	17.66	0	0.12	0	0
0.9	0	0	0.12	0	0

Table 17. Clustering performance in terms of silhouettes using the reduced dataset with PCA applied to normalized data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	25.60	17.89	10.24	12.48	5.59
0.8	11.48	14.12	9.71	4.00	2.77
0.9	0	0	0	0	0

Table 18. Clustering performance in terms of silhouettes using the reduced dataset with SPQR applied to normalized data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	27.37	15.30	9.24	0.06	3.06
0.8	14.42	10.12	0.18	0.06	0.06
0.9	0	0	0	0.06	0.06

In these experiments, the results obtained using normalized data seem to be worse than those obtained with the raw data.

8.4. TUANDROMD (Tezpur University Android Malware Dataset) Dataset

8.4.1. Clustering and Silhouettes Using Raw Data

This dataset could be considered a sort of spare matrix. There are many 0s, but there are no features containing not-a-number values (NaN). Applying PCA to this dataset, losing less than 2% of the total variation in the data, produced a dataset with seven dimensions.

The following Tables 19–21 show the obtained results in terms of silhouettes varying the number of clusters from 10 to 50 (see columns). Each row represents the percentage of points with a silhouette greater than a given threshold, that is 0.7, 0.8, and 0.9.

Table 19. Clustering performance in terms of silhouettes using original data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	38.64	36.29	36.33	12.09	36.51
0.8	38.17	21.03	21.16	11.91	21.34
0.9	17.74	17.74	17.78	0.17	17.96

Table 20. Clustering performance in terms of silhouettes using the reduced dataset with PCA.

Silhouette\N. Clusters	10	20	30	40	50
0.7	62.27	52.95	52.01	63.44	61.53
0.8	45.16	48.29	46.93	58.51	58.19
0.9	38.64	43.27	31.09	55.51	53.22

Table 21. Clustering performance in terms of silhouettes using the reduced dataset with SPQR.

Silhouette\N. Clusters	10	20	30	40	50
0.7	48.47	48.81	48.76	48.56	52.93
0.8	37.09	40.36	40.32	40.12	48.31
0.9	21.84	29.05	29.00	40.12	39.8

8.4.2. Clustering and Silhouettes Using Normalized Data

The original dataset was normalized between 0 and 1. Applying PCA to this dataset, losing less than 2% of the total variation in the data, produced a dataset with seven dimensions.

The following Tables 22–24 show the obtained results in terms of silhouettes varying the number of clusters from 10 to 50 (see columns). Each row represents the percentage of points with a silhouette greater than a given threshold, that is 0.7, 0.8, and 0.9.

Table 22. Clustering performance in terms of silhouettes using normalized data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	36.82	30.39	36.37	30.10	30.04
0.8	21.07	29.92	21.21	29.63	29.56
0.9	0.04	18.18	17.83	17.89	17.83

Table 23. Clustering performance in terms of silhouettes using the reduced dataset with PCA applied to normalized data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	62.27	58.06	58.71	58.75	58.51
0.8	45.16	52.86	53.67	54.45	54.83
0.9	38.64	34.83	45.43	49.84	50.33

Table 24. Clustering performance in terms of silhouettes using the reduced dataset with SPQR applied to normalized data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	48.16	48.90	48.76	49.66	52.91
0.8	28.33	40.45	40.32	41.21	44.46
0.9	28.33	29.14	29.01	29.90	39.85

8.5. Wine Dataset

8.5.1. Clustering and Silhouettes Using Raw Data

The original dataset is composed of 13 numerical features. Applying PCA to this dataset, losing less than 2% of the total variation in the data, produced a dataset with a single dimension. The following Tables 25–27 show the obtained results in terms of silhouettes varying the number of clusters from 10 to 50.

Table 25. Clustering performance in terms of silhouettes using original data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	62.92	57.86	42.13	40.45	42.13
0.8	45.50	38.20	25.28	24.15	28.65
0.9	10.67	7.86	6.18	10.11	14.61

Table 26. Clustering performance in terms of silhouettes using the reduced dataset with PCA.

Silhouette\N. Clusters	10	20	30	40	50
0.7	70.22	70.79	73.59	74.16	68.54
0.8	58.43	59.55	58.99	58.99	57.30
0.9	28.65	26.40	31.46	31.46	39.89

Table 27. Clustering performance in terms of silhouettes using the reduced dataset with SPQR.

Silhouette\N. Clusters	10	20	30	40	50
0.7	65.73	71.35	75.28	74.16	74.16
0.8	52.81	60.11	61.80	57.30	60.11
0.9	21.91	31.46	34.83	29.77	41.57

8.5.2. Clustering and Silhouettes Using Normalized Data

The original dataset was normalized between 0 and 1. Applying PCA to this dataset, losing less than 2% of the total variation in the data, produced a dataset with three dimensions. The following Tables 28–30 show the obtained results in terms of silhouettes varying the number of clusters from 10 to 50.

Table 28. Clustering performance in terms of silhouettes using normalized data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	0.56	1.12	1.68	1.68	1.123
0.8	0.56	0.56	1.12	1.68	0.56
0.9	0.56	0.56	1.12	1.68	0.56

Table 29. Clustering performance in terms of silhouettes using the reduced dataset with PCA applied to normalized data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	5.62	8.99	8.99	10.11	12.92
0.8	0	1.68	4.49	3.93	8.42
0.9	0	1.123	0	0	5.62

Table 30. Clustering performance in terms of silhouettes using the reduced dataset with SPQR applied to normalized data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	14.61	14.61	10.67	19.10	14.61
0.8	3.93	4.49	3.93	8.99	8.43
0.9	0	0	3.37	2.81	2.81

8.6. Multiple Features Dataset

8.6.1. Clustering and Silhouettes Using Raw Data

The original dataset is composed of 649 numerical features. Applying PCA to this dataset, losing less than 2% of the total variation in the data, produced a dataset with two dimensions. The following Tables 31–33 show the obtained results in terms of silhouettes varying the number of clusters from 10 to 50.

Table 31. Clustering performance in terms of silhouettes using original data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	22.40	2.40	0.70	0.35	0.70
0.8	15.95	0	0	0	0
0.9	0	0	0	0	0

Table 32. Clustering performance in terms of silhouettes using the reduced dataset with PCA.

Silhouette\N. Clusters	10	20	30	40	50
0.7	56.55	32.30	24.95	24.10	23.45
0.8	38.65	10.90	7.10	8.50	6.400
0.9	15.65	0	0.35	0.35	0

Table 33. Clustering performance in terms of silhouettes using the reduced dataset with SPQR.

Silhouette\N. Clusters	10	20	30	40	50
0.7	65.35	55.10	48.60	44.85	44.95
0.8	51.60	35.80	28.75	25.85	25.50
0.9	22.45	8.85	3.65	3.55	2.10

8.6.2. Clustering and Silhouettes Using Normalized Data

The original dataset was normalized between 0 and 1. Applying PCA to this dataset, losing less than 2% of the total variation in the data, produced a dataset with four dimensions. The following Tables 34–36 show the obtained results in terms of silhouettes varying the number of clusters from 10 to 50.

Table 34. Clustering performance in terms of silhouettes using normalized data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	0	0.20	0.10	0.15	0
0.8	0	0.20	0.10	0.15	0
0.9	0	0.20	0.10	0.15	0

Table 35. Clustering performance in terms of silhouettes using the reduced dataset with PCA applied to normalized data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	21.05	8.70	6.90	2	1.40
0.8	4.15	0.40	0.65	0	0
0.9	0	0	0	0	0

Table 36. Clustering performance in terms of silhouettes using the reduced dataset with SPQR applied to normalized data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	73.70	74.85	69.25	67.75	72.35
0.8	67.70	64.60	57.20	58.50	61.75
0.9	46.45	45.45	29.80	23.35	40.60

8.7. Dry Beans Dataset

8.7.1. Clustering and Silhouettes Using Raw Data

The original dataset is composed of 16 numerical features. Applying PCA to this dataset, losing less than 2% of the total variation in the data, produced a dataset with one dimension. The following Tables 37–39 show the obtained results in terms of silhouettes varying the number of clusters from 10 to 50.

Table 37. Clustering performance in terms of silhouettes using original data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	62.15	59.83	60.83	60.52	59.70
0.8	48.18	45.64	46.20	45.95	45.34
0.9	17.24	16.09	16.72	15.99	14.38

Table 38. Clustering performance in terms of silhouettes using the reduced dataset with PCA.

Silhouette\N. Clusters	10	20	30	40	50
0.7	62.20	61.43	61.27	60.45	61.16
0.8	48.29	46.52	46.95	46.86	46.85
0.9	17.46	17.48	18.41	18.37	19.23

Table 39. Clustering performance in terms of silhouettes using the reduced dataset with SPQR.

Silhouette\N. Clusters	10	20	30	40	50
0.7	62.26	61.13	61.31	61.02	60.66
0.8	48.14	47.12	47.61	46.80	46.85
0.9	17.40	17.43	18.61	18.40	18.11

8.7.2. Clustering and Silhouettes Using Normalized Data

The original dataset was normalized between 0 and 1. Applying PCA to this dataset, losing less than 2% of the total variation in the data, produces a dataset with two dimensions. The following Tables 40–42 show the obtained results in terms of silhouettes varying the number of clusters from 10 to 50.

Table 40. Clustering performance in terms of silhouettes using normalized data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	13.16	4.39	3.22	0.55	0.53
0.8	3.11	2.36	2.31	0	0
0.9	0.15	0.037	0.01	0	0

Table 41. Clustering performance in terms of silhouettes using the reduced dataset with PCA applied to normalized data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	30.07	22.94	19.43	17.23	17.13
0.8	10.10	6.74	5.55	3.91	4.74
0.9	0	0	0	0	0

Table 42. Clustering performance in terms of silhouettes using the reduced dataset with SPQR applied to normalized data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	30.55	21.61	20.99	19.95	21.64
0.8	11.39	4.94	4.33	4.63	5.99
0.9	0	0	0	0	0

8.8. APS Failure at Scania Trucks Dataset

8.8.1. Clustering and Silhouettes Using Raw Data

The original dataset is composed of 168 numerical features. Applying PCA to this dataset, losing less than 2% of the total variation in the data, produced a dataset with one dimension. The following Tables 43–45 show the obtained results in terms of silhouettes varying the number of clusters from 10 to 50.

Table 43. Clustering performance in terms of silhouettes using original data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	11.94	37.07	29.13	45.43	45.31
0.8	11.31	32.07	20.36	39.45	39.34
0.9	9.70	27.02	5.54	35.07	34.99

Table 44. Clustering performance in terms of silhouettes using the reduced dataset with PCA.

Silhouette\N. Clusters	10	20	30	40	50
0.7	68.40	74.05	78.86	75.34	77.41
0.8	58.14	65.12	70.83	66.66	68.81
0.9	41.40	41.38	51.89	46.52	50.42

Table 45. Clustering performance in terms of silhouettes using the reduced dataset with SPQR.

Silhouette\N. Clusters	10	20	30	40	50
0.7	90.66	85.51	81.28	84.09	79.63
0.8	85.30	80.79	76.01	78.15	72.96
0.9	81.17	65.34	59.48	63.08	56.15

8.8.2. Clustering and Silhouettes Using Normalized Data

The original dataset was normalized between 0 and 1. Applying PCA to this dataset, losing less than 2% of the total variation in the data, produced a dataset with five dimensions. The following Tables 46–48 show the obtained results in terms of silhouettes varying the number of clusters from 10 to 50.

Table 46. Clustering performance in terms of silhouettes using normalized data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	34.28	20.97	18.43	9.97	11.9
0.8	9.65	13.52	15.20	2.99	8.56
0.9	7.08	5.12	5.42	1.97	0.40

Table 47. Clustering performance in terms of silhouettes using the reduced dataset with PCA applied to normalized data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	48.17	25.93	21.18	16.80	12.46
0.8	18.63	15.13	14.07	8.93	4.05
0.9	8.01	2.48	1.94	0	1.16

Table 48. Clustering performance in terms of silhouettes using the reduced dataset with SPQR applied to normalized data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	28.40	27.88	14.55	12.74	12.79
0.8	15.10	11.13	9.59	6.58	7.49
0.9	6.88	3.37	4.38	2.75	3.95

8.9. Relative location of CT Slices on Axial Axis Dataset

8.9.1. Clustering and Silhouettes Using Raw Data

The original dataset is composed of 16 numerical features. Applying PCA to this dataset, losing less than 2% of the total variation in the data, produced a dataset with six dimensions. The following Tables 49–51 show the obtained results in terms of silhouettes varying the number of clusters from 10 to 50.

Table 49. Clustering performance in terms of silhouettes using original data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	0.002	0.002	0.004	0.013	0.019
0.8	0.002	0.002	0.004	0.013	0.019
0.9	0.002	0.002	0.004	0.013	0.019

Table 50. Clustering performance in terms of silhouettes using the reduced dataset with PCA.

Silhouette\N. Clusters	10	20	30	40	50
0.7	10.06	7.16	4.99	6.56	3.79
0.8	0.82	2.91	2.88	2.858	0.68
0.9	0	1.54	1.42	1.19	0

Table 51. Clustering performance in terms of silhouettes using the reduced dataset with SPQR.

Silhouette\N. Clusters	10	20	30	40	50
0.7	23.61	25.91	15.64	20.30	37.95
0.8	8.39	5.75	0.01	15.531	21.03
0.9	5.38	0.01	0.01	5.47	8.31

8.9.2. Clustering and Silhouettes Using Normalized Data

The original dataset was normalized between 0 and 1. Applying PCA to this dataset, losing less than 2% of the total variation in the data, produced a dataset with five dimensions. The following Tables 52–54 show the obtained results in terms of silhouettes varying the number of clusters from 10 to 50.

Table 52. Clustering performance in terms of silhouettes using normalized data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	0.006	0.007	0.011	0.015	0.020
0.8	0.006	0.007	0.011	0.015	0.020
0.9	0.006	0.007	0.011	0.015	0.020

Table 53. Clustering performance in terms of silhouettes using the reduced dataset with PCA applied to normalized data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	13.45	10.93	5.20	2.15	1.46
0.8	1.46	4.32	3.18	0	0
0.9	0	1.40	1.46	0	0

Table 54. Clustering performance in terms of silhouettes using the reduced dataset with SPQR applied to normalized data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	53.43	82.86	90.65	88.22	85.94
0.8	25.01	79.76	86.64	83.62	81.28
0.9	0	57.74	79.67	70.08	67.80

8.10. Mice Protein Expression Dataset

8.10.1. Clustering and Silhouettes Using Raw Data

The original dataset is composed of 81 numerical features. Applying PCA to this dataset, losing less than 2% of the total variation in the data, produced a dataset with one dimension. The following Tables 55–57 show the obtained results in terms of silhouettes varying the number of clusters from 10 to 50.

Table 55. Clustering performance in terms of silhouettes using original data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	0	0.18	0.09	0.28	0.37
0.8	0	0.18	0.09	0.28	0.37
0.9	0	0.18	0.09	0.28	0.37

Table 56. Clustering performance in terms of silhouettes using the reduced dataset with PCA.

Silhouette\N. Clusters	10	20	30	40	50
0.7	63.42	62.41	63.70	63.70	64.63
0.8	50.09	49.07	48.61	50.92	50.65
0.9	23.05	19.35	21.85	21.76	19.07

Table 57. Clustering performance in terms of silhouettes using the reduced dataset with SPQR.

Silhouette\N. Clusters	10	20	30	40	50
0.7	58.15	61.85	64.35	63.15	63.15
0.8	47.59	50.92	49.44	49.91	49.72
0.9	17.87	21.30	22.68	21.39	22.41

8.10.2. Clustering and Silhouettes Using Normalized Data

The original dataset was normalized between 0 and 1. Applying PCA to this dataset, losing less than 2% of the total variation in the data, produced a dataset with one dimension. The following Tables 58–60 show the obtained results in terms of silhouettes varying the number of clusters from 10 to 50.

Table 58. Clustering performance in terms of silhouettes using normalized data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	0	0	0	0.09	0.18
0.8	0	0	0	0.09	0.18
0.9	0	0	0	0.09	0.18

Table 59. Clustering performance in terms of silhouettes using the reduced dataset with PCA applied to normalized data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	60.09	61.85	62.41	62.68	64.44
0.8	46.30	49.44	48.70	48.98	50.65
0.9	19.26	20.74	20.46	20.37	19.81

Table 60. Clustering performance in terms of silhouettes using the reduced dataset with SPQR applied to normalized data.

Silhouette\N. Clusters	10	20	30	40	50
0.7	61.11	62.78	59.91	61.941	61.851
0.8	48.981	50.181	45.551	48.051	48.421
0.9	19.26	20.83	17.78	18.80	20.37

8.11. Synthesis

The mean values of all the results shown in the previous sections are reported in the following figures. Figures 1–3 report the mean results obtained by applying the analysis to the raw datasets, while Figures 4–6 report those obtained by applying the analysis to the normalized datasets. Each figure shows the mean results in terms of silhouettes obtained varying the number of clusters from 10 to 50.

In Figure 1, each line represents the mean percentage of points with a silhouette greater than 0.7. The blue line represents the results obtained using the raw data, the orange line represents those obtained after preprocessing with PCA, and the gray line represents those obtained after preprocessing with SPQR.

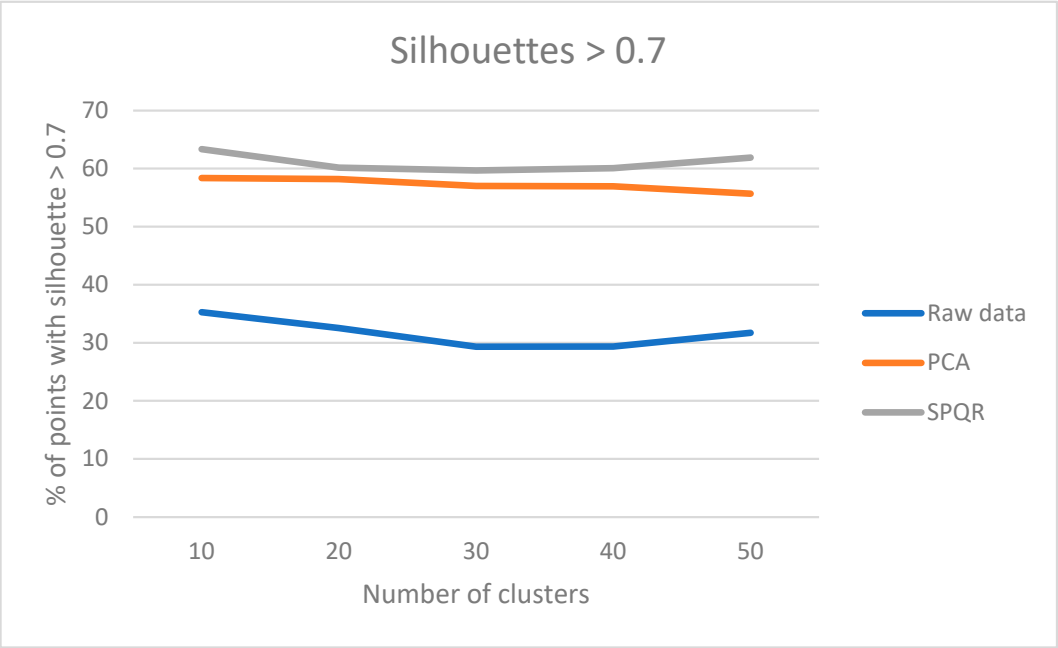

Figure 1. Mean results in terms of silhouettes >0.7 obtained by varying the number of clusters from 10 to 50.

In Figure 2, each line represents the mean percentage of points with a silhouette greater than 0.8.

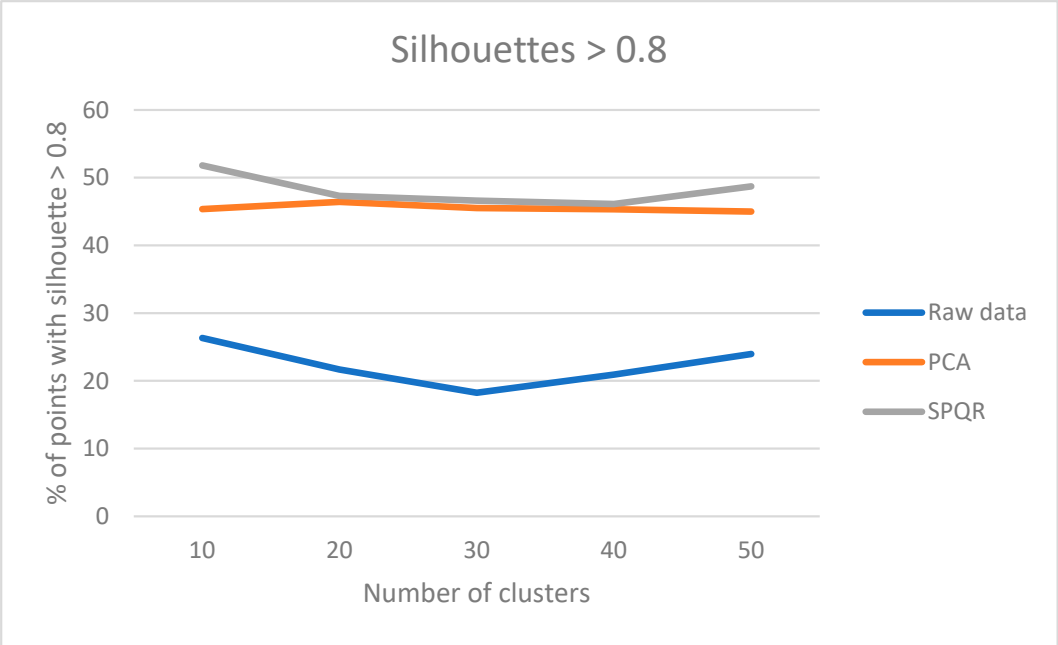

Figure 2. Mean results in terms of silhouettes >0.8 obtained by varying the number of clusters from 10 to 50.

In Figure 3, each line represents the mean percentage of points with a silhouette greater than 0.9.

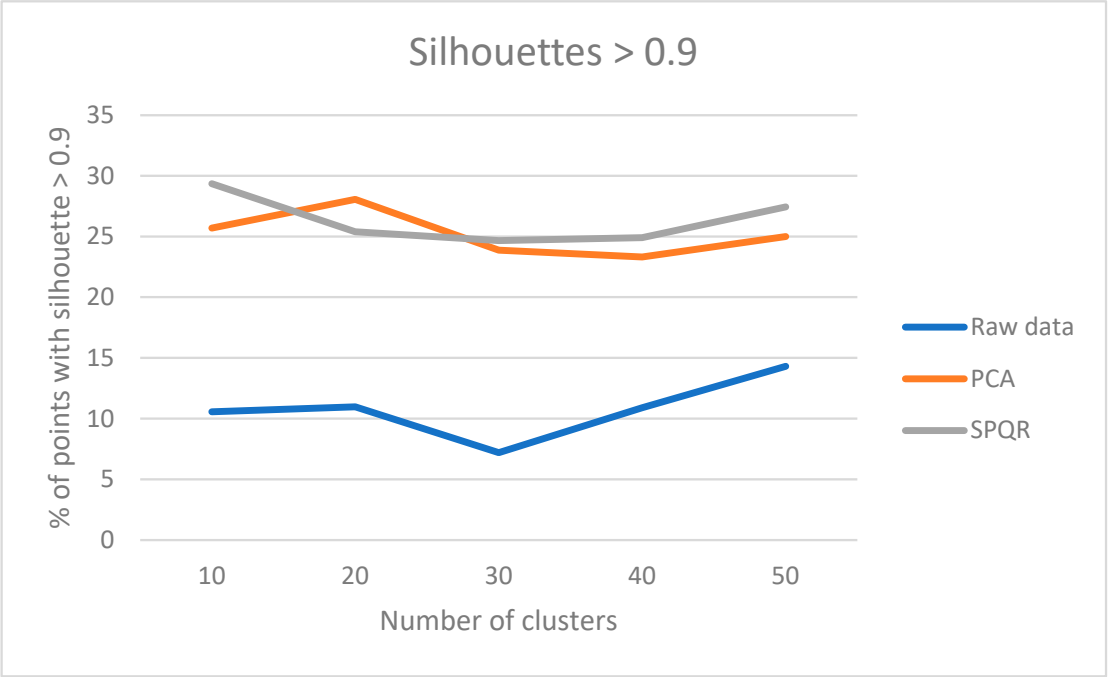

Figure 3. Mean results in terms of silhouettes >0.9 obtained by varying the number of clusters from 10 to 50.

In Figure 4, each line represents the mean percentage of points with a silhouette greater than 0.7 obtained by applying the analysis to the normalized datasets.

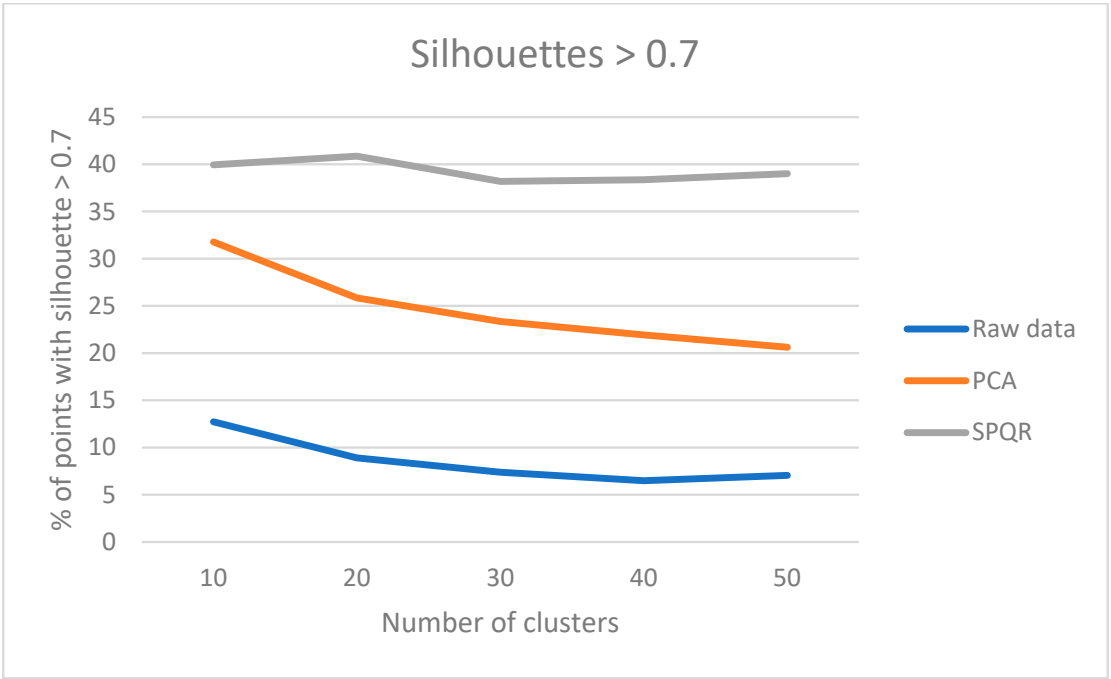

Figure 4. Mean results in terms of silhouettes >0.7 obtained by varying the number of clusters from 10 to 50 using normalized data.

In Figure 5, each line represents the mean percentage of points with a silhouette greater than 0.8 obtained by applying the analysis to the normalized datasets.

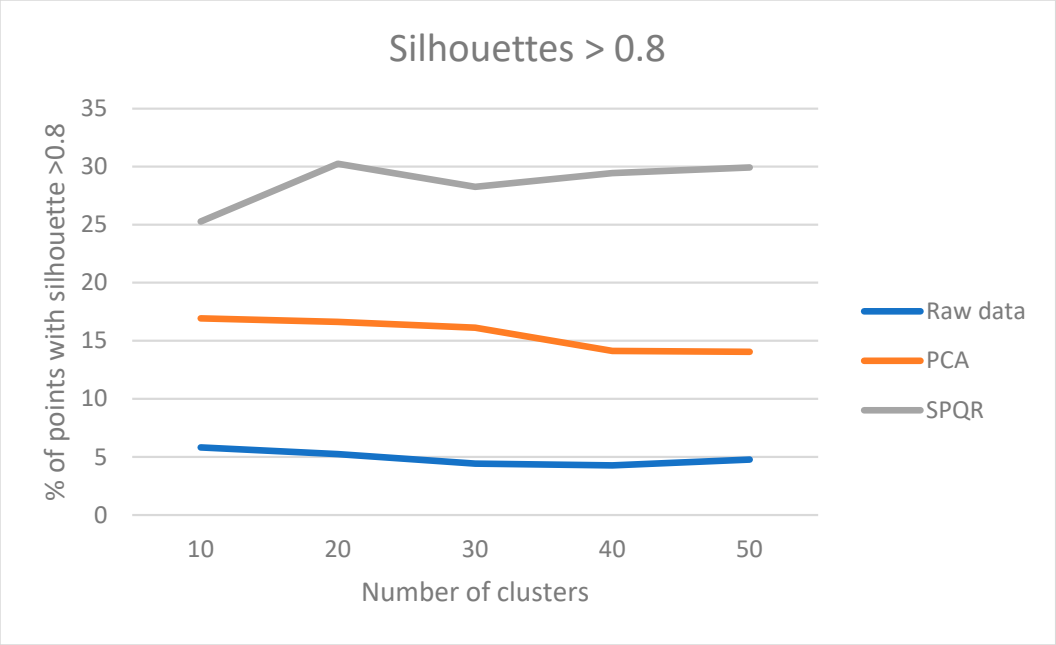

Figure 5. The mean results in terms of silhouettes >0.8 obtained varying the number of clusters from 10 to 50 using normalized data.

In Figure 6, each line represents the mean percentage of points with a silhouette greater than 0.9 obtained by applying the analysis to the normalized datasets.

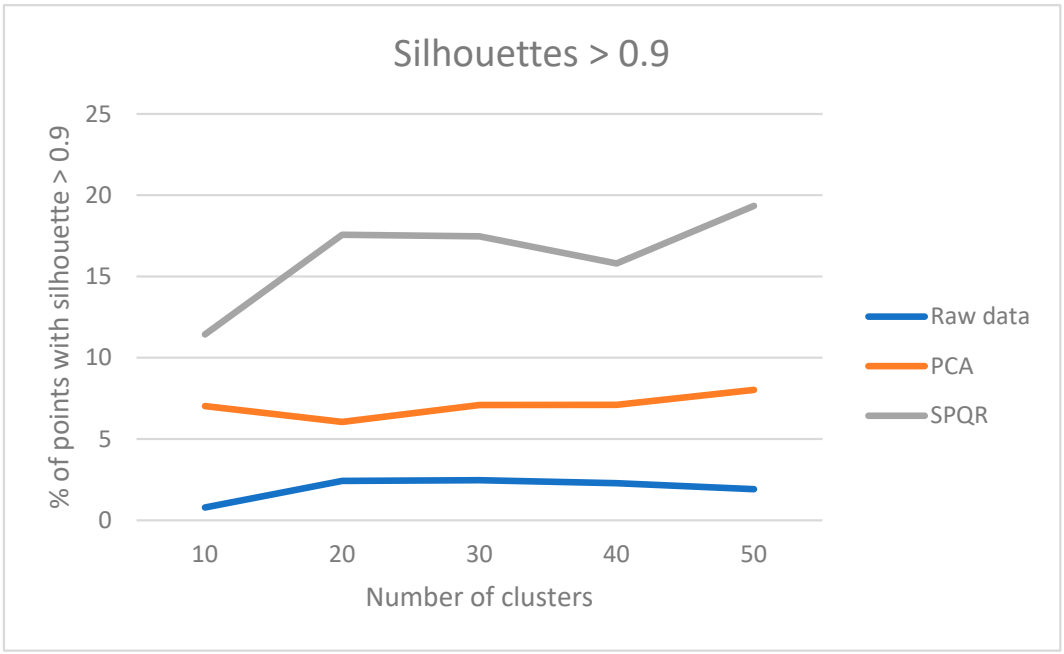

Figure 6. Mean results in terms of silhouettes >0.9 obtained by varying the number of clusters from 10 to 50 using normalized data.

To perform a comparative evaluation of various methods, statistical hypothesis tests are commonly adopted with experimental results for a number of datasets. In this work, the obtained results were analyzed using the Friedman test [44]. It is a nonparametric statistical method to evaluate the validity of the null hypothesis (no difference or relationship exists between the two sets of data or variables being analyzed). The analysis was carried out using Matlab®. Since the obtained value of χ^2 is 28.13 which is greater than the

critical value of 5.991 (derived from the table of the χ^2 distribution choosing the value of significance α equal to 0.05), we can reject the null hypothesis. Once the null hypothesis was rejected applying the Friedman test, the authors adopted a post hoc procedure for the pairwise comparison using the "*multcompare*" Matlab function. Figure 7 shows a graphical representation of the obtained results. In this figure, the intervals are computed so that the two estimates being compared are significantly different if their intervals are disjointed and are not significantly different if their intervals overlap. In Figure 7, all the intervals are disjoined, indicating the fact that the three observations are significatively different among them.

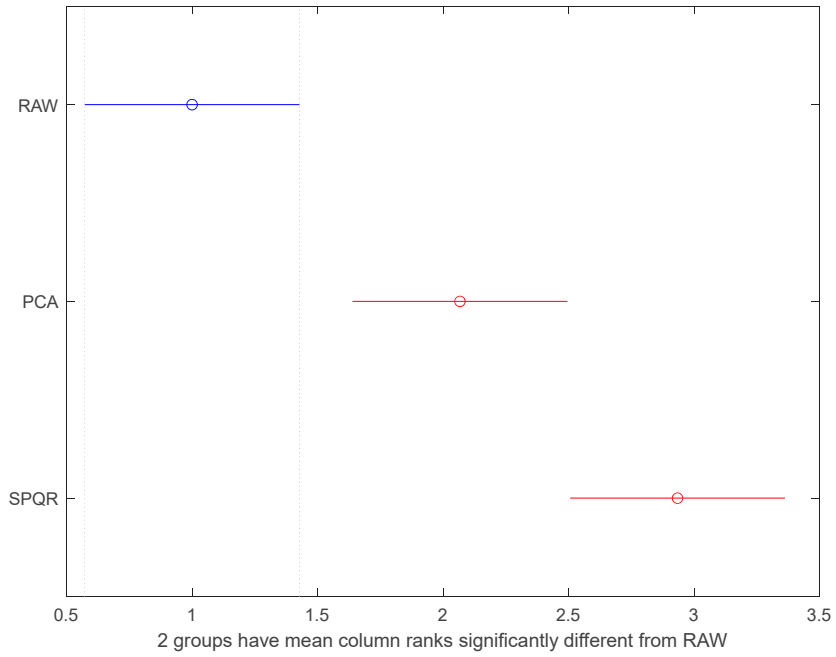

Figure 7. Results of the "multicompare" procedure.

9. Discussion

In this work, the authors proposed to use the SPQR algorithm as a data preprocessing method for machine learning algorithms. It is a technique for sparce matrix approximation, and to best of our knowledge, it has never been applied to this context. The obtained results were compared to those obtained using the well-known PCA under the same conditions. In particular, an unsupervised learning method (FCM) was applied to ten publicly available databases using different preprocessing techniques:

1. No preprocessing: in these experiments, the FCM was used to cluster raw data;
2. Data normalization: in these experiments, each feature of the datasets was normalized;
3. PCA: the dimensions of the datasets were reduced using PCA, retaining at least the 98% of the total variation in the data;
4. SPQR: the dimensions of the datasets were reduced by applying the same number of features used in PCA.

The obtained results are shown in the previous section, and they allow us to outline some considerations:

- Data normalization has a significant impact on the performance of the clustering algorithms. This is due to the "equalization effect" that data normalization has on the morphology of the feature space defined by a dataset. Indeed, when normalizing a dataset, each dimension of the feature space has the same extension (1); hence, all the dimensions of the feature space have the same weight in the distance function used in the clustering algorithm. From a semantic point of view, this situation could cause trouble when analyzing a dataset where there are features more important than others;

- Reducing the dimensions of the dataset could improve the classification performance of the algorithms. In the proposed experiments, there were improvements both in the computational and classification performances;
- As shown in Section 7, reducing dimensions using PCA means to reproject the original feature space into a new one defined by the eigenvectors of the covariance matrix. This fact makes the estimation of the contribution given by each feature of the original dataset to the classification process quite difficult. Furthermore, it is not possible to classify new points eventually added in a second time to the original dataset without recasting them into the new feature space;
- As shown in Section 6, the SPQR algorithm does not introduce any changes in the original dataset (it changes only the position of some features in the original dataset). This overcomes the drawbacks of PCA discussed above. Furthermore, the proposed results show that the performance obtained by preprocessing data with this algorithm often overcomes that obtained using PCA.

Author Contributions: Investigation, V.D.L. and A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: http://archive.ics.uci.edu/ml/index.php.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gerbrands, J.J. On the relationships between SVD, KLT and PCA. *Pattern Recognit.* **1981**, *14*, 375–381. [CrossRef]
2. Tufféry, S. *Data Mining and Statistics for Decision Making*; Wiley: Hoboken, NJ, USA, 2011.
3. Adolfo, C.M.S.; Chizari, H.; Win, T.Y.; Al-Majeed, S. Sample Reduction for Physiological Data Analysis Using Principal Component Analysis in Artificial Neural Network. *Appl. Sci.* **2021**, *11*, 8240. [CrossRef]
4. Buatoom, U.; Jamil, M.U. Improving Classification Performance with Statistically Weighted Dimensions and Dimensionality Reduction. *Appl. Sci.* **2023**, *13*, 2005. [CrossRef]
5. Stewart, G. Four algorithms for the the efficient computation of truncated pivoted QR approximations to a sparse matrix. *Numer. Math.* **1999**, *83*, 313–323. [CrossRef]
6. Popolizio, M.; Amato, A.; Piuri, V.; Di Lecce, V. Improving Classification Performance Using the Semi-pivoted QR Approximation Algorithm. In *Rising Threats in Expert Applications and Solutions: Proceedings of FICR-TEAS, Jaipur, India, 4 July 2022*; Rathore, V.S., Sharma, S.C., Tavares, J.M.R., Moreira, C., Surendiran, B., Eds.; Springer: Singapore, 2022; Volume 434, pp. 263–271. [CrossRef]
7. Dua, D.; Graff, C. *UCI Machine Learning Repository*; University of California, School of Information and Computer Science: Irvine, CA, USA, 2019; Available online: http://archive.ics.uci.edu/ml (accessed on 28 February 2023).
8. Meng, Z.; Zhang, Z.; Zhang, D.; Yang, D. An active learning method combining Kriging and accelerated chaotic single loop approach (AK-ACSLA) for reliability-based design optimization. *Comput. Methods Appl. Mech. Eng.* **2019**, *357*, 112570. [CrossRef]
9. Meng, Z.; Li, G.; Yang, D.; Zhan, L. A new directional stability transformation method of chaos control for first order reliability analysis. *Struct. Multidiscip. Optim.* **2016**, *55*, 601–612. [CrossRef]
10. de Velasco, M.; Justo, R.; Zorrilla, A.L.; Torres, M.I. Analysis of Deep Learning-Based Decision-Making in an Emotional Spontaneous Speech Task. *Appl. Sci.* **2023**, *13*, 980. [CrossRef]
11. Balasubramanian, M.; Schwartz, E.L. The Isomap Algorithm and Topological Stability. *Science* **2002**, *295*, 7. [CrossRef]
12. Roweis, S.T.; Saul, L.K. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science* **2000**, *290*, 2323–2326. [CrossRef]
13. Donoho, D.L.; Grimes, C. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proc. Natl. Acad. Sci. USA* **2003**, *100*, 5591–5596. [CrossRef]
14. Belkin, M.; Niyogi, P. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Comput.* **2003**, *15*, 1373–1396. [CrossRef]
15. Huang, H.; Feng, H. Gene Classification Using Parameter-Free Semi-Supervised Manifold Learning. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2011**, *9*, 818–827. [CrossRef] [PubMed]
16. Shawe-Taylor, J.; Cristianini, N. *Kernel Methods for Pattern Analysis*; Cambridge University Press: Cambridge, UK, 2004.
17. Giraud, C. *Introduction to High-Dimensional Statistics*; CRC Press: Boca Raton, FL, USA, 2014; Volume 138.

18. Adragni, K.P.; Al-Najjar, E.; Martin, S.; Popuri, S.K.; Raim, A.M. Group-wise sufficient dimension reduction with prin-cipal fitted components. *Comput. Statist.* **2016**, *31*, 923–941. [CrossRef]
19. Guo, Z.; Li, L.; Lu, W.; Li, B. Groupwise Dimension Reduction via Envelope Method. *J. Am. Stat. Assoc.* **2015**, *110*, 1515–1527. [CrossRef]
20. Ward, A.D.; Hamarneh, G. The Groupwise Medial Axis Transform for Fuzzy Skeletonization and Pruning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *32*, 1084–1096. [CrossRef]
21. Zhou, J.; Wu, J.; Zhu, L. Overlapped groupwise dimension reduction. *Sci. China Math.* **2016**, *59*, 2543–2560. [CrossRef]
22. Jolliffe, I. *Principal Component Analysis*; Wiley: Hoboken, NJ, USA, 2002.
23. Johnson, R.A.; Wichern, D.W. *Applied Multivariate Statistical Analysis*; Prentice: Englewood Cliffs, NJ, USA, 1992; Volume 4.
24. Fodor, I.K. A Survey of Dimension Reduction Techniques. *arXiv* **2002**, arXiv:1403.2877.
25. Brand, M. Fast low-rank modifications of the thin singular value decomposition. *Linear Algebra Its Appl.* **2005**, *415*, 20–30. [CrossRef]
26. Gui, J.; Wang, S.-L.; Lei, Y.-K. Multi-step dimensionality reduction and semi-supervised graph-based tumor classification using gene expression data. *Artif. Intell. Med.* **2010**, *50*, 181–191. [CrossRef] [PubMed]
27. Berry, M.; Pulatova, S.; Stewart, G. Computing sparse reduced-rank approximations to sparse matrices. *ACM Trans. Math. Softw.* **2005**, *31*, 252–269. [CrossRef]
28. Minguillón, J.; Meneses, J.; Aibar, E.; Ferran-Ferrer, N.; Fābregues, S. Exploring the gender gap in the Spanish Wikipedia: Differences in engagement and editing practices. *PLoS ONE* **2021**, *16*, e0246702. [CrossRef] [PubMed]
29. Borah, P.; Bhattacharyya, D.K.; Kalita, J.K. Malware Dataset Generation and Evaluation. In Proceedings of the 2020 IEEE 4th Conference on Information & Communication Technology (CICT), Chennai, India, 3–5 December 2020; IEEE: Piscataway, NJ, USA, 2020.
30. Singh, A.P.; Jain, V.; Chaudhari, S.; Kraemer, F.A.; Werner, S.; Garg, V. Machine Learning-Based Occupancy Estimation Using Multivariate Sensor Nodes. In Proceedings of the 2018 IEEE Globecom Workshops (GC Wkshps), Abu Dhabi, United Arab Emirates, 9–13 December 2018.
31. Golovenkin, S.E.; Bac, J.; Chervov, A.; Mirkes, E.M.; Orlova, Y.V.; Barillot, E.; Gorban, A.N.; Zinovyev, A. Trajectories, bifurcations, and pseudo-time in large clinical datasets: Applications to myocardial infarction and diabetes data. *Gigascience* **2020**, *9*, giaa128. [CrossRef]
32. Koklu, M.; Ozkan, I.A. Multiclass Classification of Dry Beans Using Computer Vision and Machine Learning Techniques. *Comput. Electron. Agric.* **2020**, *174*, 105507. [CrossRef]
33. Higuera, C.; Gardiner, K.J.; Cios, K.J. Self-Organizing Feature Maps Identify Proteins Critical to Learning in a Mouse Model of Down Syndrome. *PLoS ONE* **2015**, *10*, e0129126. [CrossRef] [PubMed]
34. Saxena, A.; Prasad, M.; Gupta, A.; Bharill, N.; Patel, O.P.; Tiwari, A.; Er, M.J.; Ding, W.; Lin, C.-T. A review of clustering techniques and developments. *Neurocomputing* **2017**, *267*, 664–681. [CrossRef]
35. Pedrycz, W. Algorithms of fuzzy clustering with partial supervision. *Pattern Recognit. Lett.* **1985**, *3*, 13–20. [CrossRef]
36. Dunn, J.C. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *J. Cybern.* **1973**, *3*, 32–57. [CrossRef]
37. Bezdek, J.C.; Ehrlich, R.; Full, W. FCM: The fuzzy c-means clustering algorithm. *Comput. Geosci.* **1984**, *10*, 191–203. [CrossRef]
38. Smeulders, A.; Worring, M.; Santini, S.; Gupta, A.; Jain, R. Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 1349–1380. [CrossRef]
39. Rand, W.M. Objective Criteria for the Evaluation of Clustering Methods. *J. Am. Stat. Assoc.* **1971**, *66*, 846. [CrossRef]
40. Rousseeuw, P.J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **1987**, *20*, 53–65. [CrossRef]
41. Visalakshi, S.; Radha, V. A literature review of feature selection techniques and applications: Review of feature selection in data mining. In Proceedings of the 2014 IEEE International Conference on Computational Intelligence and Computing Research, Coimbatore, India, 18–20 December 2014; pp. 1–6. [CrossRef]
42. Krömer, P.; Platoš, J.; Snasel, V. Genetic Algorithm for the Column Subset Selection Problem. In Proceedings of the 2014 Eighth International Conference on Complex, Intelligent and Software Intensive Systems (CISIS), Birmingham, UK, 2–4 July 2014; pp. 16–22. [CrossRef]
43. Stewart, G.W. Error Analysis of the Quasi-Gram–Schmidt Algorithm. *SIAM J. Matrix Anal. Appl.* **2005**, *27*, 493–506. [CrossRef]
44. Friedman, M. The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *J. Am. Stat. Assoc.* **1937**, *32*, 675–701. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.