

Effects of Dataset Sanitization
on KMeans Cluster Quality:

Experimental Evaluation Using a
Purpose-Built Unit Test Suite

A Technical Report

Prepared by:

James M. Brennan

Date:

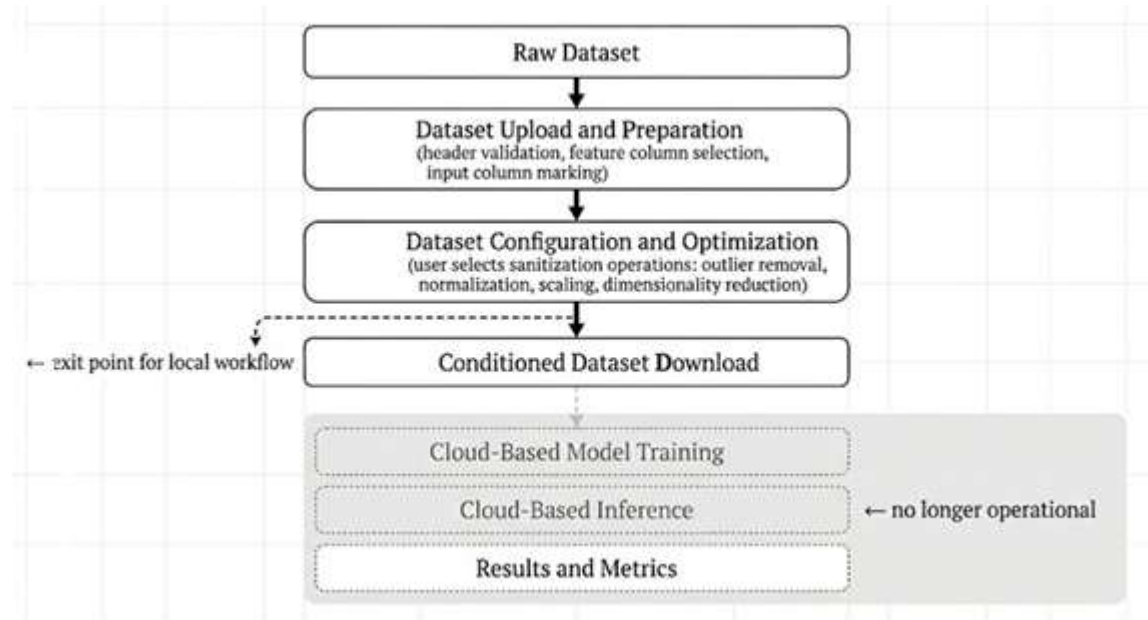
May 2026

Rev. 001

October 3, 2026

Preface

This document and its accompanying software originated from a need to complete an interrupted workflow: the overarching objective being to validate dataset configuration and optimization operations during unsupervised model training and inference. The starting point was a cloud-based no-code machine learning platform designed to guide users through the full lifecycle of a machine learning project, from initial dataset preparation through model training and inference, without requiring the user to write any code. The platform's intended workflow scenario for clustering proceeds through several stages as diagrammed below:

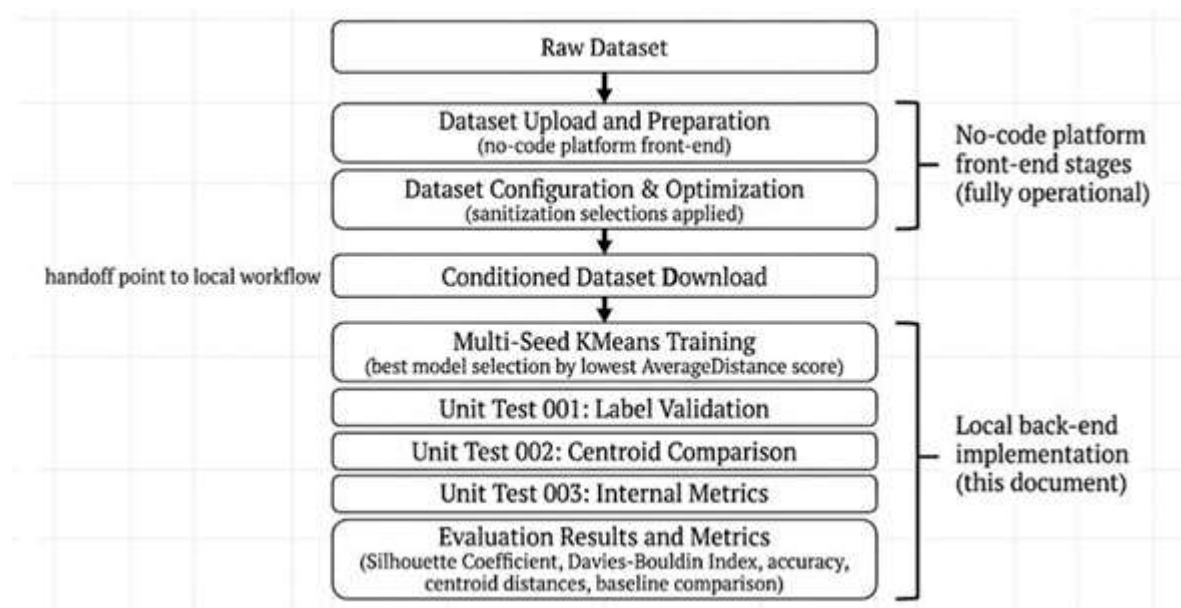


The platform's front-end stages consist of the dataset upload, feature column preparation, and Configuration and Optimization (sanitization) sections. This sanitization section offers a selection of operations that affect the dataset in some manner and has remained fully functional and accessible allowing download of the conditioned dataset. Following sanitization however, the back-end cloud services responsible for model training, inference, and results reporting became non-operational in the middle of testing by this Dev, leaving the workflow incomplete at the point of the conditioned dataset download.

Rather than abandon the work already invested in understanding and using the platform's front-end preparation capabilities, the decision was made to build a local software implementation

that picks up the workflow at the conditioned dataset download step and carry it through to completion. This implementation was focused on replicating the core “KMeans clustering operation”¹ the cloud back-end would have performed, handle the non-deterministic initialization behavior inherent in KMeans, and provide evaluation metrics capable of assessing both model quality and the effectiveness of the no-code platform's sanitization options, since understanding the value of those options was a primary motivation for using the platform in the first place.

The result of this effort is the local back-end workflow described in this report, implemented in C# (named “SeedClustering”) using Microsoft ML.NET 8.0. It accepts the conditioned dataset downloaded from the no-code platform as its primary input, applies a multi-seed KMeans training strategy to reliably identify the best performing model across multiple random initializations, and evaluates the trained model using a suite of three purpose-built unit tests. Together these components complete the intended workflow of the no-code platform entirely within a local execution environment, as illustrated below:



During development of local implementations for Unit Tests 001 and 002 it was discovered that certain sanitization choices, such as those involving feature value transformation like normalization and standardization, will alter the geometric structure of the dataset. This finding motivated the development of Unit Test 003, which uses internal cluster quality metrics requiring no ground truth labels at all, and which proved to be the only reliable evaluation method for heavily conditioned dataset variants.

¹ ML.NET currently supports a centroid-based approach using K-Means clustering [0.0.1]

It is the author's hope that the local back-end workflow described here will be useful for understanding the consequences of preprocessing decisions in unsupervised machine learning applications.

1. Introduction

1.1 Purpose of the Document

This document examines how dataset sanitization operations affect the quality of unsupervised machine learning models specifically trained using the KMeans clustering algorithm. The sanitized dataset, produced by a no-code platform, is intercepted and downloaded following the user selected sanitization process. This renamed dataset is then externally processed using the KMeans clustering algorithm. Specifically, it investigates how common data preprocessing operations such as outlier removal, and feature scaling "alters" the geometric structure of cluster datasets and then influences the ability of a trained model to correctly identify and separate natural groupings within the data.

The analysis is grounded in a practical software implementation named "SeedClustering" written in C# using Microsoft ML.NET 8.0, and supported by a suite of three Unit Tests designed to measure model quality from complementary perspectives. Together the Unit Tests provide a systematic framework for quantifying the effects of sanitization across multiple dataset variants, from the original unmodified raw data, through increasingly aggressive conditioning operations.

1.2 Scope

This document focuses specifically on unsupervised KMeans clustering applied to synthetic datasets generated using the Python scikit-learn `make_classification` library [8.1.0]. The synthetic datasets consist of 300 samples, 20 features, and 3 classes with one cluster per class, providing a controlled environment where the true cluster structure is known in advance through a companion ground truth labels file.

The sanitization operations examined are drawn from a no-code data optimization platform that offers selections across three categories: 1) Outlier removal methods including IQR filtering, Local Outlier Factor, Elliptic Envelope, Z-Score, and Isolation Forest; 2) Data normalization and scaling methods including Min-Max Normalization, RobustScaler, Standardization, MaxAbsScaler, and KernelCenterer; and 3) Dimensionality reduction methods including PCA, TSNE, and SVD. Not all of these sanitization categories are addressed by this report which only features components from category 1 and 2.

The C# implementation uses Microsoft ML.NET 8.0 for model training and inference, and employs a multi-seed training strategy to mitigate the non-deterministic behavior inherent in KMeans centroid initialization. The trained model is evaluated using a suite of three purpose-built unit tests described in detail in this report.

1.3 Overview of the Sanitization Problem

Raw datasets generated by simulation tools or collected from real-world sources commonly contain outliers that skew the learned model parameters, features measured at widely different scales that cause distance-based algorithms to weight some dimensions disproportionately. For this reason dataset sanitization, the process of cleaning, transforming, and restructuring raw data prior to training, is a standard and necessary step in any machine learning workflow.

For supervised learning problems, where labeled training examples guide the model toward known correct outputs, the effects of sanitization are relatively straightforward to evaluate. Unsupervised clustering presents a different and more challenging situation. KMeans and related algorithms learn by identifying natural geometric groupings in the feature space without reference to any labels. When sanitization operations transform that structure by removing rows, rescaling feature values, or projecting the data into a new coordinate system they “alter” the very thing the clustering algorithm is trying to model.

This creates problems two of which highlighted in the analysis presented in this report. First, outlier removal drops rows from the dataset, which desynchronizes the ground truth labels file from the remaining data. Second, feature scaling operations such as normalization and standardization transform all feature values into a new numeric range, which means the conditioned dataset and the original raw dataset no longer share the same feature space.

The combined effect of these problems is that the ground truth labels generated alongside the raw dataset progressively lose their validity as sanitization becomes more “aggressive”. This document will demonstrate how a purpose-built unit test suite addresses this by combining ground-truth-based evaluation where it remains valid with internal geometric quality metrics that require no labels at all, enabling meaningful model evaluation across the full spectrum of sanitization variants!

2. Importance of Dataset Sanitization for Unsupervised Cluster Training

2.1 What Sanitization Means in the Context of Clustering

Dataset sanitization is the process of preparing raw data before it is used to train a machine learning model. It covers a range of operations including removing erroneous data points, adjusting the scale of feature values so that no one feature dominates the others, and reducing the number of features.

In supervised learning, where the correct answer for each training example is already known, sanitization is relatively simple to evaluate. You clean the data, train the model, and check whether prediction accuracy on a test set went up or down. The labels tell you whether the sanitization helped or hindered the result.

Clustering is different because there are no labels to guide the process. A clustering algorithm such as KMeans learns entirely by looking at the geometric relationships between data points, and whether natural groupings emerge from these proximities. This makes clustering more sensitive to sanitization than supervised learning, because any operation that changes the feature values, or removes data points, changes the geometry that the algorithm is trying to learn from. This brings out the need for a strategy that can detect when sanitization has improved the data and when it has degraded the cluster structure the model is trying to learn.

2.2 Categories of Sanitization Operations and Their Effects

The no-code platform used in this report organizes its sanitization options into three sections. Each section contains operations that affect the dataset in different ways, but only a few of these options are actually used in this report: Z-Score from Section 1, and Normalization from Section 2. A review of all selectable option differences is touched upon below to illustrate their differences, suggesting further future work in these areas.

In Section 1 outlier removal operations remove rows from the dataset. The result of this is that the dataset shrinks, and the ground truth labels file becomes misaligned with the remaining rows of the conditioned file.

In Section 2 normalization and scaling operations leave the number of rows unchanged but transform the feature values into a new numeric range. The side effect is that the conditioned dataset and the original raw dataset no longer share the same feature space.

In Section 3 dimensionality reduction operations go further than scaling by replacing the original feature columns entirely with a smaller set of derived components. This makes comparison with the raw data impossible, making the ground truth labels invalid for evaluation.

2.3 Ground Truth Invalidation

The synthetic dataset used with this testing is generated with a companion ground truth labels file that identifies the true class membership of every row. This file allows the unit tests to verify whether the clustering model found the correct groupings. However, sanitization operations affect the alignment with the dataset, which can skew that alignment in two ways.

The first is row dropping. When outlier removal operations delete rows from the dataset the labels file still contains entries for those deleted rows. The two files are now out of step with each other so direct comparison between predicted cluster assignments and true labels give unreliable results.

The second is feature value transformation. When normalization or scaling operations are applied the numeric values in the dataset change, but the labels file is unchanged. The dataset now exists in a different feature space than the one the labels were generated from.

Experimental results during development illustrated this problem. When the nearest neighbor approach was used to recover proxy ground truth labels for a conditioned dataset that had been normalized and Z-scored, the Euclidean distance between each conditioned row and its closest raw counterpart averaged approximately 9.09 units, which was far from the near-zero distance that would indicate a valid match. Every conditioned row was consequently mapped to the same proxy label, reducing the three classes into one and producing a false 100% accuracy result. This finding confirmed that feature value transformation renders the ground truth labels file unusable for any form of comparison with the conditioned dataset, and motivated the adoption of internal cluster quality metrics in Unit Test 003, which evaluate the conditioned dataset entirely on its own terms without reference to the original ground truth labels.

2.4 Balancing Sanitization Benefit Against Cluster Structure Preservation

Sanitization is not inherently harmful to clustering because outlier removal can improve cluster separation by eliminating data points that sit between natural groupings and blur their boundaries. Feature scaling can improve KMeans performance by preventing features with large numeric ranges from dominating the distance calculations that drive cluster assignment. Used with careful consideration, sanitization can make the cluster structure in a dataset cleaner and easier for the algorithm to find.

The risk arises when sanitization is applied too aggressively by removing too many rows which can shrink one class disproportionately relative to the others thereby shifting the balance of the dataset and pulling cluster centroids away from their true positions. Also, by applying multiple scaling operations in combination can distort the feature space in ways that bring naturally separated clusters closer together and reducing the separation that KMeans relies on to draw clean boundaries between groups.

The experimental results presented in this report show the raw unmodified dataset, used as the baseline, produced a Silhouette Coefficient of 0.4264 and a Davies-Bouldin Index of 0.9519 when evaluated with Unit Test 003 (see Appendix C-1 for console output from this baseline run). These baseline scores represent the best achievable cluster quality for this dataset and model combination. Each conditioned variant is then compared against these baseline values, with the percentage change in both metrics serving as a direct measure of how much the chosen sanitization operations helped or hurt the cluster structure.

The central question this document aims to answer is therefore not whether to sanitize, but which sanitization choices preserve cluster integrity well enough to justify their use in the workflow.

3. Baseline Dataset Generation

3.1 Synthetic Dataset Generation Using sklearn make_classification

The dataset used in this report was generated using the `make_classification` function from the Python scikit-learn library (see Appendix E and E-1 Script). Synthetic data was chosen because it provides a known ground truth, where the true class membership of every generated sample is available at the time of generation, which allows the unit tests to verify whether the clustering model found the correct groupings.

The four key parameters used to generate the dataset are:

- **n_samples = 300** — the dataset contains 300 rows, providing a manageable but statistically meaningful sample size for a three-class clustering problem.
- **n_features = 20** — each sample is described by 20 numeric feature columns, labeled A through T in the dataset file.
- **n_classes = 3** — the data contains three distinct classes, corresponding to the three clusters the KMeans model is trained to find.
- **class_sep = 3.0** — the class separation parameter controls how far apart the three classes are positioned in feature space. A value of 3.0 produces well-separated clusters that are visually distinct when the data is projected onto a PCA plot, as shown in Appendix D.

The generated dataset is saved as a CSV file with a header row, with each of the 20 feature columns as inputs to the model. A companion true labels file containing one integer class label per row is saved separately and used as the ground truth reference for Unit Test 001 and Unit Test 002.

3.2 The True Labels Companion File

When `make_classification` generates the dataset it also produces a corresponding set of class labels, one per sample row, identifying which of the three classes each sample row belongs to. These labels are saved separately as a companion CSV file containing a single column of integer values (0, 1, or 2) aligned row-for-row with the main dataset file. Unit Test 001 uses it to verify that the cluster assignments produced by the trained model are consistent with the known true classes. Unit Test 002 uses it to compute the true class centroids from the raw data. These true

class centroids are then used as “ground truth centroids” to compare against the predicted cluster centroids in Unit Test 002.

The companion file is only reliable when used with the raw unmodified dataset, where every row in the dataset has a corresponding label entry and the two files are perfectly aligned. This alignment is progressively degraded as sanitization operations are applied. For these reasons the companion true labels file is used as a baseline reference resource rather than a universally applicable evaluation tool.

3.3 The Raw Baseline Dataset

The raw baseline dataset is the unmodified CSV file produced directly by `make_classification` with no sanitization applied. It contains all 300 samples across 20 feature columns, with its companion true labels file fully aligned row-for-row. In the context of this report it serves as the reference point against which all sanitized dataset variants are compared.

The baseline dataset plays a specific and important role in the unit test suite. It is always the first dataset processed in a test session. Unit Test 001 performs its most reliable evaluation on the baseline because the true labels file is fully aligned and no rows have been dropped. Unit Test 002 uses the baseline's raw feature values and true labels to compute the true class centroids that serve as the reference for centroid comparison (Note: if test dataset values have not been altered, otherwise the Unit Test 002 is skipped. Unit Test 003 runs against this baseline to establish the Silhouette Coefficient and Davies-Bouldin Index scores that all subsequent conditioned dataset runs are measured against.

To enforce test ordering the dataset selection popup used in the software restricts the user to selecting only the baseline dataset first in a new session as seen in figure 3.3.1.

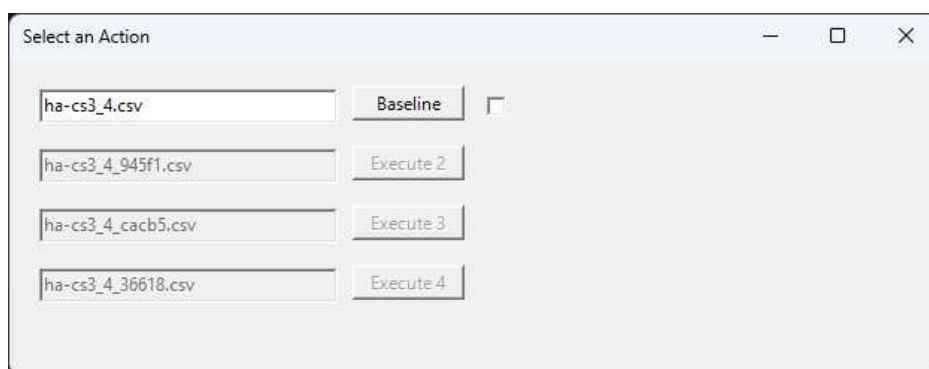


Figure 3.3.1. Dataset selection popup restricts the user to select only baseline dataset first, with other dataset's grayed out.

The buttons for conditioned dataset variants are disabled until the baseline run has been completed and the Unit Test 003 baseline scores have been stored. This design ensures that every session begins from a consistent and well-defined reference point before any conditioned variant is evaluated. It is worth noting that running the unit tests on the raw baseline dataset also serves as a validation of the trained model itself, independent of any sanitization considerations. A result of 100% accuracy in Unit Test 001, zero centroid distance in Unit Test 002, and Silhouette and Davies-Bouldin scores consistent with well-separated clusters in Unit Test 003, taken together, confirm that the multi-seed training strategy successfully identified the correct cluster structure in the data.

3.4 PCA Visualization of the Baseline Dataset

Principal Component Analysis, commonly referred to as PCA, is a mathematical technique that reduces high-dimensional data to a smaller number of derived dimensions called principal components. In this report it is used as a visualization tool to project the 20-dimensional feature space of the dataset down to two dimensions so that the cluster structure can be inspected visually. The two principal components selected are those that capture the largest proportions of the total variance in the data, making them the most informative two-dimensional view available.

For the baseline dataset used in this report the first principal component (PC1) captures 26.1% of the total variance and the second (PC2) captures 22.4%, together accounting for 48.5% of the total variance across all 20 features (see Appendix D for the PCA plot). However, the PCA visualization also introduces something to be aware of: the plot captures only 48.5% of the total variance with the remaining 51.5% distributed across the 18 dimensions not shown. In those hidden dimensions the clusters may be less cleanly separated than the plot suggests. The PCA plot is a useful first confirmation that genuine cluster structure exists in the data. The Silhouette Coefficient, discussed elsewhere, provides the more complete measurement of that structure across all 20 dimensions, and it is the Silhouette score rather than the visual impression that serves as the quantitative baseline for all subsequent comparisons in this document.

4. The Multi-Seed Training Strategy

4.1 The KMeans Initialization Problem

KMeans clustering begins by randomly placing a set of starting points, called centroids, in the feature space. Each data point is then assigned to its nearest centroid, and the centroids are iteratively moved to the geometric center of their assigned points until the assignments stop changing. This process is called convergence, and depends heavily on where the centroids were placed at the start. The final result determines which data points end up in which cluster and may experience boundary issues due to the randomly placed starting points.

Because the starting positions are chosen randomly, different runs of the KMeans algorithm on the same dataset can produce different results. In most cases the differences are small, but in some cases a poor set of starting positions can cause the algorithm to converge on a suboptimal solution. This initialization problem is a well-known limitation of the standard KMeans algorithm. This means that a single training run cannot be trusted to reliably find the best possible clustering. Running the algorithm once and then accepting that result risks producing a model whose cluster assignments are influenced by the random starting configuration rather than by the actual structure of the data. This limitation motivated the multi-seed training strategy, which addresses the initialization problem by running the KMeans algorithm multiple times with different random (seeds) starting configurations and selecting the best overall result.

4.2 The Multi-Seed Approach

The multi-seed training strategy addresses the initialization problem described in Section 4.1 by training multiple KMeans models, each starting from a different random configuration, and selecting the best performing model from the set. In the software implementation each training run is controlled by an integer seed value, which uses it to initialize all random operations within that run including the placement of the initial cluster centroids. By trying a predefined list of seed values, the strategy systematically samples a range of different starting configurations rather than relying on a single random initialization.

For each seed value in the list the software trains a complete KMeans model on the 80% training portion of the dataset, evaluates it against the held-out 20% test portion, and records the resulting “AverageDistance” score. After all seed values have been tried the model that produced the lowest “AverageDistance” score is selected as the best model and saved for use by the unit test suite [4.2.0].

The seed list used in this report contains 19 values, a set chosen to provide a broad sampling of initialization configurations without making the training phase prohibitively slow. For the baseline dataset with class separation of 3.0 this number of seeds used in this document is based upon proven methods to consistently identify a stable best model [4.2.1]. An important property of this approach is full reproducibility. Re-running the software with the same seed list and the same dataset will always produce the same best model.

4.3 Saving and Reusing the Best Model

Once the multi-seed training loop has completed and the best performing model has been identified by its lowest “AverageDistance” score, the model is saved to disk as a compressed zip file using the ML.NET model persistence API. This saved file contains not just the learned cluster centroid coordinates but the complete trained pipeline, including the feature concatenation step and the normalization parameters learned from the training data during the fitting process.

The practical consequence for the unit test suite is that all three unit tests load and use the same saved model file, guaranteeing that every evaluation in a session is performed against an identical model. Rerunning the software is sufficient to trigger a completely fresh training cycle with a new best model selection.

5. Unit Test 001 Label Validation Against Ground Truth

5.1 Unit Test 001 Purpose and Scope

Unit Test 001 validates that the cluster assignments produced by the trained KMeans model are consistent with the known true class labels provided in the companion true labels file. It is the most direct evaluation available in the test suite, where it compares the model's predicted cluster IDs against ground truth on a row-by-row basis and reports an overall accuracy score.

Unit Test 001 occupies a specific and complementary role within the three-test suite. It is the only test that performs explicit row-level ground truth comparison, making it the most sensitive indicator of whether the model found the correct class partition. A result of 100% accuracy on the raw baseline dataset, as achieved in the experimental runs described in Section 8, is the strongest possible confirmation that the multi-seed training strategy identified the correct clustering solution.

5.2 Label Mapping

A direct numeric comparison between the predicted cluster IDs produced by KMeans and the true class labels in the companion file is not possible without a preliminary mapping step. This is because KMeans assigns arbitrary integer identifiers to the clusters it finds (using ML.NET these are the values 1, 2, and 3) with no relationship to the true class labels in the companion file, which are the values 0, 1, and 2 generated by scikit-learn.

To resolve this issue Unit Test 001 software uses a majority-vote label mapping operation which is constructed before any row-by-row comparison is made. For each distinct true label value this mapping identifies which predicted cluster ID appears most frequently among the data points that carry that true label. The true label is then mapped to that cluster. Once the mapping is established it is checked that the mapping is unambiguous (all true labels map to different predicted clusters) otherwise it is flagged as ambiguous and a test failed warning is issued.

5.3 Test Methodology

Unit Test 001 follows a straightforward sequence of operations that includes a row-by-row accuracy comparison between predicted cluster assignments and mapped true labels.

The test first loads the true labels from the companion file. The number of labels to use for testing is determined from the size of the true labels file, taking 20% of the total row count up to a maximum of 60 rows and matches the test fraction used during training

The saved model is then loaded from disk and used to run inference on the selected dataset. The total row count of the loaded dataset is compared against the row count of the true labels file and any difference detected. This mismatch signals the user that a conditioned dataset with dropped rows is being evaluated and that the accuracy result should be interpreted with caution.

When inference is completed label mapping is constructed and checked for ambiguity. The mapped predicted labels are then compared to the true labels row by row across the full test set and the comparison results are displayed in a condensed table showing the first six and last six rows are being evaluated. The overall correct count and accuracy percentage are computed from the full set of comparisons and reported alongside the label mapping used, providing a complete and traceable record of how the accuracy result was reached.

Row	True Label	Predicted	Mapped	Match
1	2	2	2	YES
2	0	1	1	YES
3	0	1	1	YES
4	2	2	2	YES
5	1	3	3	YES
6	1	3	3	YES
- - -				
55	0	1	1	YES
56	2	2	2	YES
57	2	2	2	YES
58	1	3	3	YES
59	2	2	2	YES
60	0	1	1	YES
Correct: 60 / 60				
Accuracy: 100.0%				
Threshold: 70.0%				
Label Mapping Used: {0->1, 1->3, 2->2}				

Figure 5.3.0. Mapped vs. Predicted labels with Correct count and accuracy percentage.

5.4 Metrics Produced

Unit Test 001 produces three outputs that together give a complete picture of model performance against the ground truth labels.

Each row of the table shows the true label, the raw predicted cluster ID, the mapped cluster ID after applying the label mapping, and a YES or NO match indicator

The second output is the overall accuracy score, expressed as the number of correct matches divided by the total number of test rows and reported as a percentage. On the raw baseline dataset a well-trained model with clearly separated clusters should achieve an accuracy higher than this threshold, as demonstrated by the 100% accuracy result reported in the experimental runs documented in Appendix C-1.

The third output is the label mapping used to produce the comparison, reported as a dictionary showing which true label was mapped to which predicted cluster ID. Recording this mapping explicitly in the output makes the accuracy result fully traceable and allows the user to verify that the mapping is acceptable before accepting the accuracy score.

5.5 Pass/Fail Criteria

Unit Test 001 applies two independent criteria to determine whether the test passes or fails, both of which must be satisfied simultaneously for a PASS result to be declared.

The first criterion is that the overall accuracy score must meet or exceed the configurable accuracy threshold. The default threshold is set to 70%. This threshold was chosen to allow for a small degree of misclassification in datasets where cluster boundaries are not perfectly sharp.

The second criterion is that the label mapping must be unambiguous. As described earlier an unambiguous mapping is one where each of the three true class labels maps to a different predicted cluster ID. If the mapping is ambiguous, meaning two true labels compete for the same predicted cluster, the test fails regardless of the accuracy score.

When both criteria are satisfied the test reports a PASS result and, if running on the raw baseline dataset, proceeds to launch the PCA visualization script to produce the cluster plot shown in Appendix D. When either criterion is not satisfied the test reports a FAIL result.

5.6 Limitations on Conditioned Datasets

Unit Test 001 was designed with the raw baseline dataset as its primary use case, and its reliability degrades progressively as more aggressive sanitization operations are applied to the dataset.

When only outlier removal has been applied the row count mismatch between the dataset and the true labels file is detected and reported as a warning. In this case the test continues to run but the accuracy result is unreliable.

When feature value transformation has been applied through normalization or scaling operations the situation is more complicated. After transformation the dataset occupies a different feature space than the raw reference. For this reason Unit Test 001 results on conditioned datasets should always be considered alongside the results of Unit Test 003

5.7 Unit Test 001 Contributions to the Test Suite

Despite its limitations on conditioned datasets Unit Test 001 provides something that neither Unit Test 002 nor Unit Test 003 can offer: a direct and explicit row-level confirmation that the model found the correct class partition as defined by the ground truth labels.

6. Unit Test 002 Centroid Comparison Against True Class Centers

6.1 Unit Test 002 Purpose and Scope

Unit Test 002 validates that the cluster centroids found by the trained KMeans model correctly match the true class centroids computed from the raw baseline dataset and its companion true labels file. Where Unit Test 001 determines whether the model assigned individual data points to the correct clusters, Unit Test 002 determines whether the model found the correct geometric center of each cluster in the 20-dimensional feature space.

The centroid comparison approach used in Unit Test 002 compares cluster centers rather than individual row assignments so it does not require the dataset and the true labels file to be aligned row-for-row. This means Unit Test 002 can produce results even when outlier removal has dropped rows from the dataset.

However, Unit Test 002 shares one important limitation with Unit Test 001. When the selected dataset has been processed by feature value transformation operations the raw dataset no longer shares the same feature space with it. Computing centroids from the conditioned dataset and raw dataset then becomes a comparison across incompatible numeric scales.

Outlier removal operations from Section 1 of the no-code platform, such as Z-Score outlier detection, drop rows without modifying the feature values of the remaining rows. In this case the centroid comparison remains valid because the feature space is unchanged as shown by comparing the first two data rows in figure 6.1.1.

ha-cs3_4.csv First Row Raw Data

A	B	C	D		Q	R	S	T
-1.13203	-0.94859	2.35976	4.68359	...	-1.82185	-4.38902	2.87964	0.08263

ha-cs3_4_cacb5.csv First Row Z-Score Data

A	B	C	D		Q	R	S	T
-1.13203	-0.94859	2.35976	4.68359	...	-1.82185	-4.38902	2.87964	0.08263

Figure 6.1.1. Euclidean distance check to determine if feature values have been transformed.

Unit Test 002 therefore uses a first-row Euclidean distance check to if the first row of the selected dataset matches the first row of the raw dataset within a small tolerance, and if feature values are confirmed unchanged the test proceeds. If a significant distance is detected the test skips itself with an explanatory message identifying the likely cause.

6.2 The Centroid Comparison Approach

Centroid comparison is a powerful validation technique because it treats a cluster as a single geometric entity rather than a collection of individual points and represents the distribution's "center of gravity".

When a dataset is sanitized the data still defines the same general region within the 20-dimensional feature space. Because the centroid represents the location of the group rather than the identity of individual members, the validation remains robust even when the sanitized dataset has fewer rows than the reference dataset. Centroid comparison remains valid because you can still calculate a mean for the remaining rows and determine how far that new mean has drifted from the original reference point.

6.3 Test Methodology

Unit Test 002 first loads the companion true labels file and the raw dataset, verifying that their row counts match exactly. This match is a prerequisite for computing reliable true class centroids. If the counts do not match the test fails immediately with an explanatory message.

The selected conditioned dataset is then loaded independently. Its row count is compared against the raw dataset row count and any difference is reported as the number of rows dropped by the sanitization process. The first-row Euclidean distance is then performed to determine whether the feature values have been transformed. If transformation is detected the test skips itself at this point with a message explaining the reason. If the feature values are confirmed unchanged the test proceeds.

With both datasets loaded and validated the saved model is loaded from disk and inference is run on all rows of the selected dataset, producing a predicted cluster ID for each row. The true class centroids are then computed by grouping the raw dataset rows by their true label and calculating the mean feature vector of each group across all 20 feature dimensions. The predicted cluster centroids are computed in the same way by grouping the selected dataset rows by their predicted cluster ID.

The centroid matching step then finds the best correspondence between true class centroids and predicted cluster centroids and assigns each true class to its nearest predicted cluster. The resulting mapping is checked for ambiguity then the class distribution comparison is performed

by calculating the proportion of data points in the raw dataset and comparing it against the proportion of data points assigned to the corresponding predicted with the percentage difference reported for each class.

6.4 Metrics Produced

Unit Test 002 produces three sets of outputs that together shows how well the trained model's cluster geometry matches the true class structure of the raw dataset.

The first is the centroid matching table shown in figure 6.4.1, which reports the Euclidean distance between each true class centroid and its nearest predicted cluster centroid in the 20-dimensional feature space. The table shows the true class, the predicted cluster it was matched to, the distance between them, the configured distance threshold, and a pass or warn status for each pairing. An average centroid distance across all three classes is also reported.

True Class	Best Match	Distance	Threshold	Status
0	3	0.5177	2.0000	OK
1	1	0.6528	2.0000	OK
2	2	0.3614	2.0000	OK
Centroid mapping is unambiguous. OK				
Average centroid distance: 0.5106				

Figure 6.4.1. Euclidean distance between true class and nearest predicted cluster.

The second output is the centroid mapping ambiguity check shown in figure 6.4.2, which reports whether the three true classes mapped to three distinct predicted clusters or whether two true classes competed for the same predicted cluster. An unambiguous mapping is reported as a single confirmation line. An ambiguous mapping triggers a warning advising the user that the cluster geometry may be poorly defined or that the model may not have found three genuinely distinct groupings in the data.

```
Centroid mapping is unambiguous. OK
Average centroid distance: 0.5106
```

Figure 6.4.2. Centroid mapping ambiguity check.

The third output is the class distribution comparison table shown in figure 6.4.3, which shows for each true class the count and percentage of samples in the raw dataset alongside the count and percentage of samples assigned to the corresponding predicted cluster in the selected

dataset. The absolute percentage difference between the two is reported for each class alongside a pass or warn status based on the configured tolerance threshold of 15%.

Class	True N	True%	Pred N	Pred%	Diff%	Status
0	100	33.3%	91	34.3%	1.0%	OK
1	100	33.3%	83	31.3%	2.0%	OK
2	100	33.3%	91	34.3%	1.0%	OK
All classes within 15% distribution tolerance. OK						

Figure 6.4.3. Count and percentage assigned in raw and selected dataset.

6.5 Pass/Fail Criteria

Unit Test 002 applies two independent criteria to determine whether the test passes or fails, both of which must be satisfied simultaneously for a PASS result to be declared.

The first is that the centroid mapping must be unambiguous. An unambiguous mapping is one where each of the three true class centroids maps to a different predicted cluster centroid. An ambiguous mapping causes an immediate FAIL regardless of how close centroid distances are.

The second criterion is that the class distribution comparison must show that the proportion of data points in each predicted cluster is within the 15% tolerance of the corresponding true class proportion. A distribution difference exceeding 15% for any class triggers a WARN status for that class and contributes to an overall FAIL result.

The centroid distance table and its threshold column are provided as diagnostic information to help the user assess the quality of the centroid matching, but the pass/fail decision rests solely on the ambiguity check and the distribution comparison.

```
=====
UT-002 RESULT: PASS for selected file ha-cs3_4_cacb5.csv
  Centroid mapping:          unambiguous
  Average centroid distance: 0.5106
  Class distribution:        within tolerance
=====
```

Figure 6.5.1. PASS RESULT indicates a PASS meeting mapping distance and tolerance.

6.6 What Unit Test 002 Uniquely Contributes to the Test Suite

Unit Test 002 has a unique contribution in directly measuring whether the trained model found the correct geometric center of each cluster in the 20-dimensional feature space, rather than evaluating individual row assignments or internal cluster geometry in isolation.

The class distribution comparison table produced by Unit Test 002 also provides information that the other two tests do not. By comparing the proportion of samples in each true class against the proportion assigned to each predicted cluster it reveals whether sanitization introduced class imbalance. This is relevant for outlier removal operations that would affect inference on new data, even if the model's internal metrics appear acceptable.

7. Unit Test 003 and Internal Cluster Quality Metrics

7.1 Unit Test 003 Purpose and Scope

Unit Test 003 evaluates intrinsic cluster quality using two internal metrics, the Silhouette Coefficient and the Davies-Bouldin Index, that require no ground truth labels. It is the only test in the suite valid for all dataset variants including those heavily conditioned by feature value transformation or dimensionality reduction operations that completely invalidate the companion true labels file.

The test measures two complementary aspects of cluster quality. The Silhouette Coefficient measures how well each individual data point fits within its assigned cluster relative to neighboring clusters. The Davies-Bouldin Index measures the ratio of within-cluster scatter to between-cluster separation, providing a view of how compact and well-separated the clusters are as a whole.

Unit Test 003 also establishes baseline Silhouette and Davies-Bouldin scores from the raw unmodified dataset on the first run of each session, then reports the percentage change for all subsequent conditioned runs. This measures how much each sanitization variant degraded or preserved the cluster structure relative to the best achievable result on that dataset.

7.2 Internal Metrics Are Necessary

Development failures in Unit Tests 001 and 002 established that an evaluation approach that relies on comparison between the conditioned and raw feature spaces is unreliable once feature value transformation has been applied. The only remaining option was to evaluate the conditioned dataset entirely on internal means by measuring the quality of the clusters found in the conditioned feature space without any reference to the original raw data or its ground truth labels. This is what the Silhouette Coefficient and Davies-Bouldin Index provide, and it is why Unit Test 003 is an essential component of the test suite.

7.3 The Silhouette Coefficient

The Silhouette Coefficient [7.3.0] is a measure of how well each individual data point fits within its assigned cluster relative to the neighboring clusters [7.3.1]. It provides a point-level perspective of cluster membership quality that can be aggregated to produce both per-cluster and overall scores for the entire dataset.

For each data point “i” in the dataset the Silhouette score is computed from two distance measurements. The first, denoted $a(i)$, is the average Euclidean distance from point i to all other points in its own assigned cluster. A small value of $a(i)$ indicates that the point sits close to the center of its cluster and is a good fit for that cluster. The second, denoted $b(i)$, is the average Euclidean distance from point i to all points in the nearest cluster, which is the cluster other than its own that it is closest to. A large value of $b(i)$ indicates that the point is well separated from the neighboring clusters. The Silhouette score for point i is then computed as:

$$s(i) = (b(i) - a(i)) / \max(a(i), b(i))$$

This formula produces a score in the range -1 to +1. A score near +1 means the point is much closer to the center of its own cluster than to any other cluster, indicating a strong and confident cluster assignment. A score near 0 means the point sits approximately on the boundary between two clusters. A score near -1 means the point is closer to a neighboring cluster than to its own, suggesting it may have been assigned to the wrong cluster.

The overall Silhouette Coefficient for the dataset is the average of the individual point scores across all data points. Per-cluster average scores are also computed by averaging the point scores within each cluster separately, allowing the user to identify whether one particular cluster is less well-defined than the others.

An important characteristic of the Silhouette Coefficient in the context of this report is its sensitivity to the dimensionality of the feature space. Because the score is based on Euclidean distances computed across all 20 feature dimensions simultaneously, it captures the full geometric complexity of the data rather than the simplified two-dimensional view provided by the PCA plot.

7.4 The Davies-Bouldin Index

The Davies-Bouldin Index is a measure of cluster quality that evaluates the ratio of within-cluster scatter to between-cluster separation at the cluster level [7.4.0]. Where the Silhouette Coefficient provides a point-level view of how well individual data points fit their assigned clusters, the Davies-Bouldin Index provides a structural view of how compact and well-separated the clusters are as geometric objects in the feature space. Lower scores are better, with a score of zero representing the theoretical ideal of perfectly compact clusters with infinite separation between them.

For each cluster i the index first computes the average intra-cluster scatter, denoted $s(i)$, which is the average Euclidean distance from each point in the cluster to its centroid. A small value of

$s(i)$ indicates a tight and compact cluster. The index then considers every possible pair of clusters i and j and computes a similarity ratio $R(ij)$ defined as: [7.4.1]

$$R(ij) = (s(i) + s(j)) / d(\text{centroid}_i, \text{centroid}_j)$$

where $d(\text{centroid}_i, \text{centroid}_j)$ is the Euclidean distance between the two cluster centroids. A high value of $R(ij)$ indicates that two clusters are either individually scattered, close together, or both, and represent poor cluster separation. For each cluster i the worst-case similarity ratio across all other clusters is identified, and the Davies-Bouldin Index is the average of these worst-case ratios across all clusters:

$$DB = (1/k) * \sum \text{over } i \text{ of } \max(R(ij)) \text{ for } j \neq i$$

where k is the number of clusters. This formulation means the index is particularly sensitive to the worst-performing cluster pair in the dataset, making it a conservative measure of overall cluster quality.

The interpretation thresholds used in Unit Test 003 classify Davies-Bouldin scores as Excellent at or below 0.50, Good between 0.50 and 1.00, Fair between 1.00 and 2.00, and Poor above 2.00. The raw baseline dataset achieved a Davies-Bouldin Index of 0.9519, placing it in the Good range and confirming acceptable cluster separation.

7.5 The Baseline Establishment and Comparison Mechanism

The challenge in interpreting internal cluster quality metrics is that their absolute values are sensitive to characteristics of the dataset being evaluated. This includes its dimensionality and the degree of separation between its clusters. Note that a Silhouette score of 0.4264 means something very different for a 20-dimensional dataset with moderate cluster separation than it would for a two-dimensional dataset with well-separated clusters. Without a reference point it is difficult to determine if a given score represents good, acceptable, or poor cluster quality for the specific dataset and model under evaluation.

Unit Test 003 addresses this challenge through a baseline establishment and comparison operation that transforms the internal metric scores into directly interpretable measures of sanitization impact. This process operates across multiple dataset runs that occur within a single program session and depends on the user popup selector enforcing baseline-first ordering (by initially graying out all other datasets).

On the first run of each session, when the raw baseline dataset is selected (see figure 3.3.1), Unit Test 003 computes the Silhouette Coefficient and Davies-Bouldin Index for that dataset and stores their scores as the session baseline. These stored values represent the best achievable

cluster quality for this particular dataset and model combination that exists before any sanitization has been applied.

On each subsequent run, when a conditioned dataset variant is selected, Unit Test 003 computes the same two metrics for the conditioned dataset and then retrieves the stored baseline values to produce a comparison table. It also provides a status indicator that flags whether a change exceeds the configured degradation thresholds. The Silhouette Coefficient is allowed to drop by up to 15% from baseline before a warning is issued, and the Davies-Bouldin Index is allowed to increase by up to 20% before a warning is issued.

This comparison operation makes it possible to answer an important question in quantitative terms for each conditioned dataset variant: whether the cluster quality is acceptable in absolute terms, and how much the chosen sanitization operations have degraded the cluster structure relative to the unmodified baseline.

7.6 Metrics Produced

Unit Test 003 produces a richer set of outputs than either Unit Test 001 or Unit Test 002, reflecting its role as the primary evaluation tool for conditioned dataset variants where ground truth comparison is no longer available. The outputs are organized into seven groups that progress from basic cluster composition through to the final baseline comparison and pass/fail determination.

The first output is the cluster size distribution seen in figure 7.6.1, which reports the number and percentage of data points assigned to each of the three predicted clusters. This provides an immediate check on whether the model produced reasonably balanced clusters or whether one cluster has absorbed a disproportionate share of the data points, which would suggest a poorly initialized or poorly converged training run.

```
Cluster size distribution:
Cluster 1: 83 rows (31.3%)
Cluster 2: 91 rows (34.3%)
Cluster 3: 91 rows (34.3%)
```

Figure 7.6.1. number and percentage of data points assigned to each of the three predicted clusters.

The second output is the inter-centroid distance table seen in figure 7.6.2, which reports the Euclidean distance between every pair of cluster centroids in the 20-dimensional feature space. Large inter-centroid distances relative to the intra-cluster scatter values reported in the third output indicate good cluster separation. Comparing the inter-centroid distances of a

conditioned dataset run against those of the baseline run gives an immediate visual indication of whether sanitization has moved the cluster centers closer together.

Inter-centroid distances (in 20D space):	
Cluster Pair	Distance
1 <-> 2	18.3858
1 <-> 3	17.3739
2 <-> 3	17.6906

Figure 7.6.2. Inter-centroid distance table of Euclidean distance between every pair of cluster centroids in the 20-dimensional feature space.

The third output is the per-cluster intra-cluster scatter table seen in 7.6.3, which reports the average Euclidean distance from each data point to its assigned cluster centroid. This is the $s(i)$ value used in the Davies-Bouldin Index calculation described in Section 7.4 and provides a direct measure of how compact each cluster is. A large increase in scatter relative to the baseline run indicates that sanitization has spread the data points within each cluster further from their centers, degrading cluster compactness.

Per-cluster intra-cluster scatter (in 20D space):	
Cluster	Avg Dist to Centroid
1	7.6376
2	8.1227
3	8.6014

Figure 7.6.3. average Euclidean distance from each data point to its assigned cluster centroid.

The fourth output is the Davies-Bouldin Index seen in figure 7.6.4, reported as a single overall score accompanied by its interpretation label and pass/fail status. The per-cluster scatter values that contribute to this score are shown in the third output, allowing the user to identify which cluster or cluster pair is driving a poor Davies-Bouldin result if one is observed.

```
Davies-Bouldin Index: 0.9418 (Good)
Pass threshold:      <= 1.5000
Davies-Bouldin status: OK
```

Figure 7.6.4. Davies-Bouldin Index, reported as a single overall score accompanied by its interpretation label and pass/fail status.

The fifth output is the per-cluster and overall Silhouette score seen in figure 7.6.5. Each cluster's average Silhouette score is reported individually alongside an interpretation label ranging from Strong through Reasonable, Acceptable, Weak, and Poor. The overall Silhouette score is the

average across all data points and is the primary metric used for the Silhouette pass/fail determination. Comparing per-cluster scores between the baseline and conditioned runs can reveal whether sanitization degraded all three clusters equally or whether one particular cluster was more affected than the others.

Per-cluster Silhouette scores (in 20D space):		
Cluster	Avg Silhouette	Interpretation
1	0.4683	Acceptable
2	0.4356	Acceptable
3	0.3913	Acceptable
Overall Silhouette Score: 0.4306 (Acceptable)		
Pass threshold: >= 0.3000		
Silhouette status: OK		

Figure 7.6.5. Per-cluster and overall Silhouette scores.

The sixth output is the metric summary table seen in 7.6.6, which consolidates the Silhouette Coefficient and Davies-Bouldin Index results into a single formatted table showing the score, threshold, direction of improvement, interpretation label, and pass/fail status for each metric side by side. This table is designed to give the user a concise at-a-glance summary of the two primary quality indicators without having to scan back through the detailed outputs above it.

Metric	Score	Threshold	Direction	Interpretation	Status
Silhouette Coefficient	0.4306	>= 0.3	higher=better	Acceptable	OK
Davies-Bouldin Index	0.9418	<= 1.5	lower=better	Good	OK
Silhouette Coefficient interpretation (measured in 20D space):					
>= 0.70 Strong - clusters well separated					
>= 0.50 Reasonable - clusters mostly separated					
>= 0.35 Acceptable - reasonable for high-dimensional data					
>= 0.25 Weak - clusters somewhat overlapping					
< 0.25 Poor - clusters heavily overlapping					
Davies-Bouldin Index interpretation:					
<= 0.50 Excellent - very compact and well-separated					
<= 1.00 Good - acceptable cluster separation					
<= 2.00 Fair - clusters somewhat overlapping					
> 2.00 Poor - clusters heavily overlapping					

Figure 7.6.6. Metric summary table.

The seventh and final output is the baseline comparison table as seen in figure 7.6.7, which appears on all runs after the first and shows the baseline score, current score, and percentage change for both metrics alongside degradation status indicators. This table is the most directly actionable output in the entire test suite, translating the internal metric scores into a quantitative answer to the question of how much each sanitization variant affected cluster quality relative to the unmodified baseline.

Baseline dataset: ha-cs3_4.csv					
Current dataset: ha-cs3_4_cacb5.csv					
	Metric	Baseline	Current	Change	Status
	Silhouette Coefficient	0.4264	0.4306	1.0%	OK
	Davies-Bouldin Index	0.9519	0.9418	-1.1%	OK
Conditioning degradation is within acceptable limits. OK					

Figure 7.6.7. Baseline score, current score, and percentage change for both metrics alongside degradation status indicators.

7.7 Pass/Fail Criteria

Unit Test 003 applies two independent threshold criteria to determine whether the test passes or fails, both of which must be satisfied simultaneously for a PASS result to be declared. Unlike Unit Test 001 and Unit Test 002, which rely on ground truth alignment for their primary criteria, both criteria in Unit Test 003 are based entirely on the internal metric scores computed from the conditioned dataset itself, making them valid and applicable regardless of which sanitization operations were applied.

The first criterion is that the overall Silhouette Coefficient must meet or exceed the configured threshold of 0.35. This threshold was calibrated specifically for the 20-dimensional feature space of the dataset used in this work, where the raw baseline dataset with a correct model achieved a Silhouette score of 0.4264. The threshold of 0.35 provides a margin of approximately 18% below the baseline score.

The second criterion is that the Davies-Bouldin Index must not exceed the configured threshold of 1.00. This threshold corresponds to the boundary between the Good and Fair interpretation ranges and was confirmed as appropriate for this dataset by the raw baseline score of 0.9519, which fell just within the Good range.

On comparison runs against the baseline an additional set of degradation checks is applied alongside the absolute threshold criteria. The Silhouette Coefficient is flagged with a warning if it has dropped by more than 15% from its baseline value, and the Davies-Bouldin Index is flagged with a warning if it has increased by more than 20% from its baseline value. These degradation warnings do not directly affect the PASS or FAIL outcome, which is determined solely by the absolute thresholds. All four threshold and degradation values are defined as configurable static fields at the top of the Unit Test 003 class and can be adjusted to reflect the specific tolerance requirements of different datasets or applications.

7.8 What Unit Test 003 Uniquely Contributes to the Test Suite

Unit Test 003 occupies an important position in the test suite because it is the only test that can evaluate model quality across all sanitization variants tested, including those where ground truth comparison has been completely invalidated by feature value transformation

The baseline comparison operation is the most useful capability that Unit Test 003 brings to the suite. By expressing cluster quality as a percentage change from a verified baseline rather than as an absolute score, it transforms two metrics that are inherently difficult to interpret in isolation into a direct and quantitative answer to the central question answered by this report: how much did each sanitization variant degrade the cluster structure relative to the unmodified

data. This capability is not available from any other component of the test suite and is what makes Unit Test 003 the primary tool for the comparative analysis presented in this section.

The per-cluster breakdown of both the Silhouette Coefficient and the Davies-Bouldin scatter values provides a further diagnostic capability that is unique to Unit Test 003 that allows the user to see whether all three clusters were affected equally by a sanitization operation or whether one particular cluster was degraded more than the others.

8. Experimental Results and Discussion

8.1 Baseline Raw Dataset Results

The baseline raw dataset used in this document is `ha-cs3_4.csv`, and was generated using the scikit-learn `make_classification` function [8.1.0] using the parameters described in Section 3.1. It contains 300 samples across 20 feature columns with three equally balanced classes of 100 samples each, accompanied by the companion true labels file `ha-cs3_4_true_labels.csv`. All three unit tests were run against this dataset on the first run of each test session, establishing both the ground truth verification results and the internal metric baseline scores against which all subsequent conditioned dataset runs are compared.

Unit Test 001 Results

Unit Test 001 achieved 100% accuracy on the raw baseline dataset, correctly matching all 60 test rows to their true class labels after applying the label mapping. The label mapping was unambiguous, with true class mappings to predicted clusters as expected (see figure 7.6.2). This result confirms that the multi-seed training strategy successfully identified the correct class partition and that the best model saved to disk is a reliable starting point for all subsequent evaluations. The full console output for this run is documented in Appendix C-1.

Unit Test 002 Results

Unit Test 002 confirmed that the trained model found the correct geometric cluster centers with an average centroid distance of exactly 0.0000 across all three classes. The centroid mapping was unambiguous and the class distribution comparison showed a 0.0% difference between the true class proportions and the predicted cluster proportions for all three classes., reflecting the perfectly balanced 100 samples per class composition of the raw dataset.

Unit Test 003 Results

Unit Test 003 established the session baseline scores from the raw dataset run. The overall Silhouette Coefficient was 0.4264, with per-cluster scores of 0.4359 for cluster 1, 0.4526 for cluster 2, and 0.3908 for cluster 3, all falling in the Weak to Acceptable interpretation range. The Davies-Bouldin Index was 0.9519, placing it in the Good interpretation range just below the pass threshold of 1.00. Both metrics passed their respective thresholds, producing an overall PASS result for Unit Test 003 on the baseline dataset. These scores were stored as the session baseline and served as the reference point for all subsequent conditioned dataset runs documented in Section 8.2.

PCA Visualization

Following the PASS result from Unit Test 001 the Python PCA visualization script was automatically launched, producing the cluster plot shown in Appendix D. The plot confirms three distinct and well-separated cluster groupings in the two-dimensional PCA projection, with PC1 capturing 26.1% and PC2 capturing 22.4% of the total variance. As discussed earlier the visual separation apparent in the PCA plot is consistent with but not directly reflected in the Silhouette score, which measures separation across all 20 feature dimensions rather than just the two shown in the plot.

8.2 Results for Each Conditioned Dataset Variant

Variant 1 Normalization Only: ha-cs3_4_945f1.csv Min-Max Normalization was applied with no outlier removal, retaining all 300 rows. The first-row distance of 14.5605 confirmed feature value transformation, causing Unit Test 002 to skip.

Unit Test 001 achieved 100% accuracy confirming the correct class partition was found despite rescaling. Unit Test 003 reported a Silhouette of 0.3270 and Davies-Bouldin of 1.2374, both passing their thresholds, with a Davies-Bouldin degradation warning of 30.0% indicating measurable but acceptable cluster quality reduction. Result: PASS with warning.

Variant 2 Z-Score Outlier Removal: ha-cs3_4_cacb5.csv Z-Score outlier detection removed 35 rows leaving 265, with no feature value transformation confirmed by a first-row distance of 0.0000.

Unit Test 001 achieved 100% accuracy. Unit Test 002 ran fully, reporting an average centroid displacement of 0.5106 and class distribution differences of 1-2%, confirming no significant class imbalance. Unit Test 003 reported a Silhouette of 0.4306 and Davies-Bouldin of 0.9418, both marginally better than the baseline, indicating the removed rows were genuinely atypical samples. Result: PASS with no warnings.

Variant 3 Standardization with Outlier Removal: ha-cs3_4_36618.csv Standardization combined with outlier removal reduced the dataset to 265 rows. The first-row distance of 14.5385 confirmed feature value transformation, causing Unit Test 002 to skip.

Unit Test 001 achieved 100% accuracy. Unit Test 003 reported a Silhouette of 0.3152 and Davies-Bouldin of 1.2675, the weakest scores of all variants, with a Davies-Bouldin degradation warning of 33.2% reflecting the compounding effect of both row removal and feature rescaling. Result: PASS with warning.

8.3 Comparative Summary Table Across All Four Variants

The results from the four experimental runs are consolidated in the table 8.3.1 below, providing a side-by-side comparison of all key metrics across the raw baseline and three conditioned variants. Full console output for each run is documented in Appendix C-1 through C-4.

Dataset Variant	Sanitization Applied	Rows	UT-001 Accuracy	UT-002 Avg Centroid Distance	UT-003 Silhouette	UT-003 Davies-Bouldin	Verdict
ha-cs3_4.csv	None (Baseline)	300	100.0%	0.0000	0.4264	0.9519	BEST
ha-cs3_4_945f1.csv	Normalization only	300	100.0%	SKIPPED*	0.3270	1.2374	PASS/WARN
ha-cs3_4_cacb5.csv	Z-Score outlier removal	265	100.0%	0.5106	0.4306	0.9418	PASS
ha-cs3_4_36618.csv	Standardization + outlier removal	265	100.0%	SKIPPED*	0.3152	1.2675	PASS/WARN

Table 8.3.1. Comparison of all key metrics referencing raw baseline and three dataset variants.

* UT-002 skipped due to feature value transformation detected by first-row distance check

Several observations stand out from this consolidated view.

Unit Test 001 achieved 100% accuracy across all four variants including both value-transformed ones, confirming the multi-seed strategy reliably found the correct class partition regardless of sanitization applied.

Unit Test 002 ran only for the raw baseline and the Z-Score outlier removal variant. The centroid displacement of 0.5106 for the Z-Score variant quantifies the small shift in cluster centers caused by removing 35 rows. The first-row distance check correctly triggered the skip for both value-transformed variants.

Unit Test 003 showed that Z-Score outlier removal was the best performing conditioned variant, marginally improving both metrics above the baseline. Both value-transformed variants produced similar degraded scores, with Standardization combined with outlier removal performing marginally worse than Normalization alone. Both triggered the Davies-Bouldin degradation warning confirming that feature value transformation has a more significant negative effect on cluster structure than row-only outlier removal.

Best seed values varied across all four runs with seeds 5, 7, 456, and 11111 respectively confirming that each conditioned dataset presents a different geometric landscape requiring a different optimal initialization, and that a fixed single seed would have produced suboptimal results for at least some variants.

8.4 Discussion of Findings

The experimental results presented in Sections 8.1 through 8.3 reveal several findings that are worth examining in depth, both for what they confirm about expected behavior and for what they reveal about the less obvious consequences of sanitization choices.

Finding 1: KMeans is Remarkably Robust to Feature Value Transformation

The most surprising result in the experimental data is that Unit Test 001 achieved 100% accuracy on all four dataset variants, including the two where feature value transformation was applied. This was not guaranteed in advance. This suggests that for the dataset used, with its well-separated clusters and class separation parameter of 3.0, the relative geometric relationships between the three classes are preserved well enough by both normalization methods for KMeans to reliably identify the correct partition. This finding should not be generalized since datasets with lower class separation or more overlapping clusters may show greater sensitivity to feature value transformation [8.4.1].

Finding 2: Outlier Removal Preserved and Slightly Improved Cluster Quality

The Z-Score outlier removal variant produced internal metric scores that marginally exceeded the baseline values, with a Silhouette improvement of 1.0% and a Davies-Bouldin improvement of 1.1%. This result supports the interpretation that the 35 rows removed by the Z-Score outlier detector were genuinely atypical samples rather than being representative members of their respective classes. Their removal tightened the cluster structure slightly, producing a marginally cleaner geometric separation between the three groups. This is the ideal outcome of outlier removal and confirms that the Z-Score detection threshold used by the no-code platform was appropriately calibrated for this dataset.

Finding 3: Feature Value Transformation Degraded Cluster Quality Measurably

Both value-transformed variants showed meaningful degradation in the internal metrics relative to the baseline. The Silhouette scores dropped by 23.3% and 26.1% for the Normalization and Standardization variants respectively, and the Davies-Bouldin scores increased by 30.0% and 33.2%. Both Davies-Bouldin increases exceeded the 20% degradation warning threshold, triggering warnings in Unit Test 003. The source of this degradation can be analyzed by examining the inter-centroid distance data. In the raw baseline the inter-centroid distances were approximately 18 units. After normalization, or standardization, they contracted to approximately 1.05 to 1.16 units. This contraction reflects the compression of feature values into a narrower numeric range, which brings the cluster centers physically closer together in the transformed feature space even though the relative ordering of distances is preserved. The intra-cluster scatter contracted proportionally but not enough to maintain the same ratio of scatter to separation, resulting in a higher Davies-Bouldin score and a lower Silhouette score.

Finding 4: Standardization Combined with Outlier Removal Was the Worst Performing Variant

The Standardization variant produced the lowest Silhouette score of 0.3152 and the highest Davies-Bouldin score of 1.2675 of all four variants. The combination of feature value transformation and outlier removal appears to compound the negative effects of each operation individually. The outlier removal reduces the dataset size to 265 rows, which reduces the number of data points contributing to each cluster centroid and potentially makes the centroids less stable. The Standardization then rescales the remaining feature values in a way that compresses the inter-centroid distances. The combination of a smaller dataset and a compressed feature space produced the most degraded cluster geometry of all variants tested, though both metrics still passed their absolute thresholds and the overall Unit Test 003 result remained PASS.

Finding 5: The Multi-Seed Strategy Adapts to Different Dataset Geometries

The best seed value varied substantially across the four runs where seeds 5, 7, 456, and 11111 were identified as optimal for the baseline, Normalization, Z-Score, and Standardization variants respectively. This variation confirms that the optimal KMeans initialization depends on the specific geometric structure of the dataset being trained on, which changes with each sanitization variant. A fixed seed approach would have produced suboptimal models for at least some variants, potentially masking the true cluster structure and producing misleading evaluation results. The multi-seed strategy with 19 candidate seeds proved sufficient to consistently find a stable and well-performing model for all four variants tested.

Finding 6: The Three Unit Tests Provided Complementary and Non-Redundant Evidence

Across all four variants the three unit tests collectively provided information that separately they could not. Unit Test 001 provided the only direct row-level ground truth confirmation that the model found the correct class partition. Unit Test 002 provided the only direct measurement of centroid displacement caused by outlier removal, quantifying the 0.5106 unit shift in cluster centers produced by the Z-Score variant. Unit Test 003 provided the only reliable evaluation for the value-transformed variants and the only cross-variant baseline comparison that quantified the relative impact of each sanitization choice. No single test alone would have provided a complete picture of model quality across the full range of sanitization variants tested.

9. Conclusions and Recommendations

9.1 Summary of Key Findings

This document set out to investigate how dataset sanitization operations affect the quality of unsupervised KMeans clustering models trained on synthetic three-class datasets. The investigation was conducted using a local back-end software implementation built in C# with Microsoft ML.NET 8.0, which completed the workflow of a cloud-based no-code machine learning platform using a KMeans model whose back-end services had become unavailable. Four dataset variants were evaluated, a raw unmodified baseline and three conditioned variants produced by the no-code platform's Dataset Configuration and Optimization section, using a purpose-built suite of three unit tests designed to measure model quality from complementary perspectives.

The key findings from the experimental work are as follows.

Feature value transformation operations including Min-Max Normalization and Standardization preserved the correct class partition as verified by 100% accuracy in Unit Test 001 across all variants. However, they measurably degraded the internal cluster geometry as measured by Unit Test 003, with Silhouette scores dropping by 23 to 26 percent and Davies-Bouldin scores increasing by 30 to 33 percent from the baseline. Both degradation levels exceeded the configured warning threshold of 20%, indicating that while the model still found the correct clusters the geometric quality of those clusters in the transformed feature space was reduced.

Outlier removal by Z-Score detection without feature value transformation was the most beneficial sanitization operation tested. It reduced the dataset from 300 to 265 rows while marginally improving both internal metrics relative to the baseline. This variant was also the only conditioned dataset for which Unit Test 002 could run its centroid comparison, reporting an average centroid displacement of 0.5106 units from the true class centers, a small but measurable shift caused by the removal of 35 rows.

The ground truth invalidation problem identified in Section 2.3 was confirmed experimentally. Feature value transformation rendered the companion true labels file unusable for direct comparison with conditioned dataset feature values, as demonstrated by first-row Euclidean distances of approximately 14.5 units between raw and transformed rows. This finding validated the decision to develop Unit Test 003 as a ground-truth-free evaluation tool based on internal cluster quality metrics, which proved to be the only reliable evaluation method for value-transformed variants.

The multi-seed KMeans training strategy successfully identified the best performing model for each dataset variant, with the optimal seed varying substantially across variants using seeds 5, 7, 456, and 11111 for the four variants respectively. This variation confirms that a fixed single seed approach would have produced suboptimal models for at least some variants and that the multi-seed strategy is an essential component of a reliable unsupervised training workflow.

The three unit tests together provided complementary and non-redundant evidence of model quality across the full spectrum of sanitization variants. Unit Test 001 provided row-level ground truth confirmation. Unit Test 002 extended ground truth based evaluation to row-drop-only variants through centroid comparison. Unit Test 003 provided internal metric evaluation valid for all variants and a baseline comparison mechanism that quantified the relative impact of each sanitization choice. No single test alone would have been sufficient to characterize model quality across all four variants tested.

9.2 Recommended Sanitization Strategy for Cluster Datasets

The experimental results presented in this report provide a basis for practical recommendations about which sanitization operations are most compatible with unsupervised KMeans clustering and which require careful evaluation before adoption. It should be noted that not all available no-code platform sanitization operations were evaluated in this report. The recommendations below are organized in order of increasing risk to cluster structure integrity.

Recommendation 1: Prefer Outlier Removal Without Feature Value Transformation

Of all the sanitization operations tested, Z-Score outlier detection applied without any accompanying feature scaling produced the best outcome. It was the only conditioned variant that improved rather than degraded both internal metrics relative to the baseline, and it was the only variant for which the full three-test suite could be run including the centroid comparison in Unit Test 002. For datasets where outliers are a genuine concern this approach represents the lowest-risk sanitization strategy available from the no-code platform's Section 1 options. The key requirement is that the outlier removal operation is used in isolation without combining it with any Section 2 scaling operation.

Recommendation 2: Use Feature Value Transformation Only When Necessary

Min-Max Normalization and Standardization both produced measurable degradation in cluster quality as measured by Unit Test 003, with Davies-Bouldin increases of 30 to 33 percent from the baseline triggering degradation warnings in both cases. These operations should therefore be applied only when there is a specific and justified reason to do so, such as when the raw feature columns have substantially different numeric ranges that cause some features to

dominate the KMeans distance calculations. For the synthetic dataset used in this current testing, which was generated with consistent feature scales, the normalization did not improve cluster quality and only introduced geometric distortion.

When feature value transformation is unavoidable the results suggest that Min-Max Normalization and Standardization produce broadly similar outcomes for this dataset type, with Normalization performing marginally better on both metrics. Either can be used when necessary, but the Davies-Bouldin degradation warning should be treated as a signal to verify that the transformation is genuinely needed rather than applied by default.

Recommendation 3: Avoid Combining Feature Value Transformation with Outlier Removal

The Standardization variant that combined feature scaling with outlier removal produced the worst internal metric scores of all variants tested, with the lowest Silhouette score of 0.3152 and the highest Davies-Bouldin score of 1.2675. While both values still passed their absolute thresholds for this well-separated dataset, the combination of a reduced row count and a compressed feature space represents the most aggressive possible degradation of cluster geometry short of dimensionality reduction. For datasets with lower class separation than the 3.0 used here this combination could push the internal metrics below acceptable thresholds and produce unreliable cluster assignments.

Recommendation 4: Always Establish a Baseline Before Evaluating Conditioned Variants

The baseline comparison mechanism in Unit Test 003 was essential for interpreting the internal metric scores of the conditioned variants in context. Without the baseline the Silhouette score of 0.3270 for the Normalization variant would be difficult to judge as it falls in the Weak interpretation range and might be dismissed as poor performance. In the context of the baseline score of 0.4264 it is recognizable as a degraded but still functional cluster structure. The raw baseline dataset with its companion true labels file should always be the first dataset evaluated in any session, both to establish the internal metric reference values and to independently verify model correctness through Unit Test 001 and Unit Test 002 before any conditioned variants are assessed.

Recommendation 5: Increase the Seed List for Datasets With Lower Class Separation

The 19-seed list used in this work proved sufficient for a dataset with class separation of 3.0, which produces well-separated clusters. For datasets with lower class separation values such as 1.0 or 2.0, where cluster boundaries are less distinct, the KMeans initialization problem described in Section 4.1 becomes more significant and a larger seed list may be needed to reliably find the optimal model. As a practical starting point the seed list should be expanded until the best AverageDistance score stabilizes across multiple sessions, indicating that the optimal solution has been consistently found.

9.3 Limitations of This Study

The findings and recommendations presented in this document are grounded in a specific and controlled experimental setup, and several important limitations should be considered before applying them more broadly.

Limitation 1 — Results Are Specific to Synthetic Data With High Class Separation

All experimental results were obtained using synthetic datasets generated by scikit-learn's `make_classification` function with a class separation parameter of 3.0. This parameter produces well-separated clusters that are visually distinct in PCA projection and amenable to reliable KMeans identification. Real-world datasets rarely have this degree of natural separation between classes, and the favorable outcomes observed here, particularly the 100% Unit Test 001 accuracy maintained across all sanitization variants, should not be expected to generalize to datasets with lower class separation or more complex cluster geometries.

Limitation 2 — Only a Subset of Available Sanitization Operations Were Tested

The no-code platform offers a much wider range of sanitization options than the three tested in this work. From Section 1 alone, Local Outlier Factor, Elliptic Envelope, and Isolation Forest were not evaluated. From Section 2, RobustScaler, MaxAbsScaler, and KernelCenterer were not tested. Section 3 dimensionality reduction operations including PCA, TSNE, and SVD were not evaluated at all, though their expected behavior giving complete invalidation of ground truth labels and fundamental restructuring of the feature space was discussed in Section 2.2 as being “impossible”. The findings presented here apply specifically to the three operations tested and should not be assumed to extend to the untested options without further experimental validation.

Limitation 3 — The Three-Class Assumption Is Embedded Throughout

The entire unit test suite was designed around the assumption of exactly three clusters, reflecting the three-class structure of the synthetic dataset. The `NumClasses` constant, the confusion matrix dimensions, the label mapping logic, and the pass/fail criteria in all three unit tests are all calibrated for a three-class problem. Applying this software to datasets with a different number of natural clusters would require modifications to each unit test class as well as recalibration of all threshold values. The findings about sanitization effects on cluster quality may also differ for problems with more or fewer than three classes.

Limitation 4 — The Nearest Neighbor Proxy Label Approach Was Not Fully Explored

The nearest neighbor proxy labeling approach explored during the development of Unit Test 003 was abandoned after it failed completely for value-transformed datasets. For row-drop-only datasets where feature values are unchanged this approach would produce exact proxy labels, since each sanitized row would find itself as its own nearest neighbor in the raw dataset with a distance of exactly zero. A hybrid evaluation strategy that uses nearest neighbor proxy labels for row-drop-only variants and internal metrics for value-transformed variants could provide more complete evaluation coverage than the current suite achieves. This was considered but not implemented in the interest of simplicity and reliability.

Limitation 5 — Internal Metric Thresholds Were Calibrated for One Specific Dataset

The Silhouette pass threshold of 0.30 and the Davies-Bouldin pass threshold of 1.50 used in Unit Test 003 were calibrated based on the scores observed for the raw baseline dataset used in this report. Different datasets with different dimensionality, feature distributions, or natural cluster separation will produce different baseline scores, and the thresholds that are appropriate for one dataset may be too strict or too lenient for another. The recommendation to always establish a baseline from the raw dataset before evaluating conditioned variants partially addresses this limitation, but the absolute threshold values should be reviewed and recalibrated whenever the unit test suite is applied to a substantially different dataset.

9.4 Suggested Future Work

The experimental work presented in this document established a functional and well-validated evaluation framework for a specific and controlled dataset configuration. Several directions for future work would extend the scope and applicability of the findings.

Future Work 1: Evaluate the Effect of Varying Class Separation

The most useful extension of this work would be to repeat the full experimental sequence using datasets generated with lower class separation values. The scikit-learn `make_classification` `class_sep` parameter could be systematically reduced from the 3.0 used here through intermediate values down to 1.0 and below, with the complete unit test suite run against each configuration. This would establish at what point each sanitization operation begins to produce unacceptable cluster degradation and would provide a more information about the relationship between natural cluster separation and sanitization tolerance.

Future Work 2: Evaluate the Remaining Sanitization Operations

The no-code platform offers several sanitization operations that were not evaluated in this work. The operations from Section 1 use different mathematical approaches to outlier detection and may remove different subsets of rows than the Z-Score method, potentially producing different effects on cluster geometry. Operations from Section 2 scale feature values differently than Min-Max Normalization and Standardization and may preserve cluster separation better or worse depending on the distribution of the feature values. Evaluating each of these operations individually and in combination would provide a more complete characterization of the no-code platform's sanitization capabilities and their consequences for clustering model quality.

Future Work 3 — Evaluate Section 3 Dimensionality Reduction Operations

PCA, TSNE, and SVD from Section 3 of the no-code platform were discussed but not experimentally evaluated in this work. These operations are fundamentally different from the scaling and outlier removal operations in Sections 1 and 2 because they replace the original feature columns with derived components, making any comparison with the raw feature space impossible. This section should not be considered for future work at this time.

Future Work 4: Extend the Suite to Datasets With More Than Three Classes

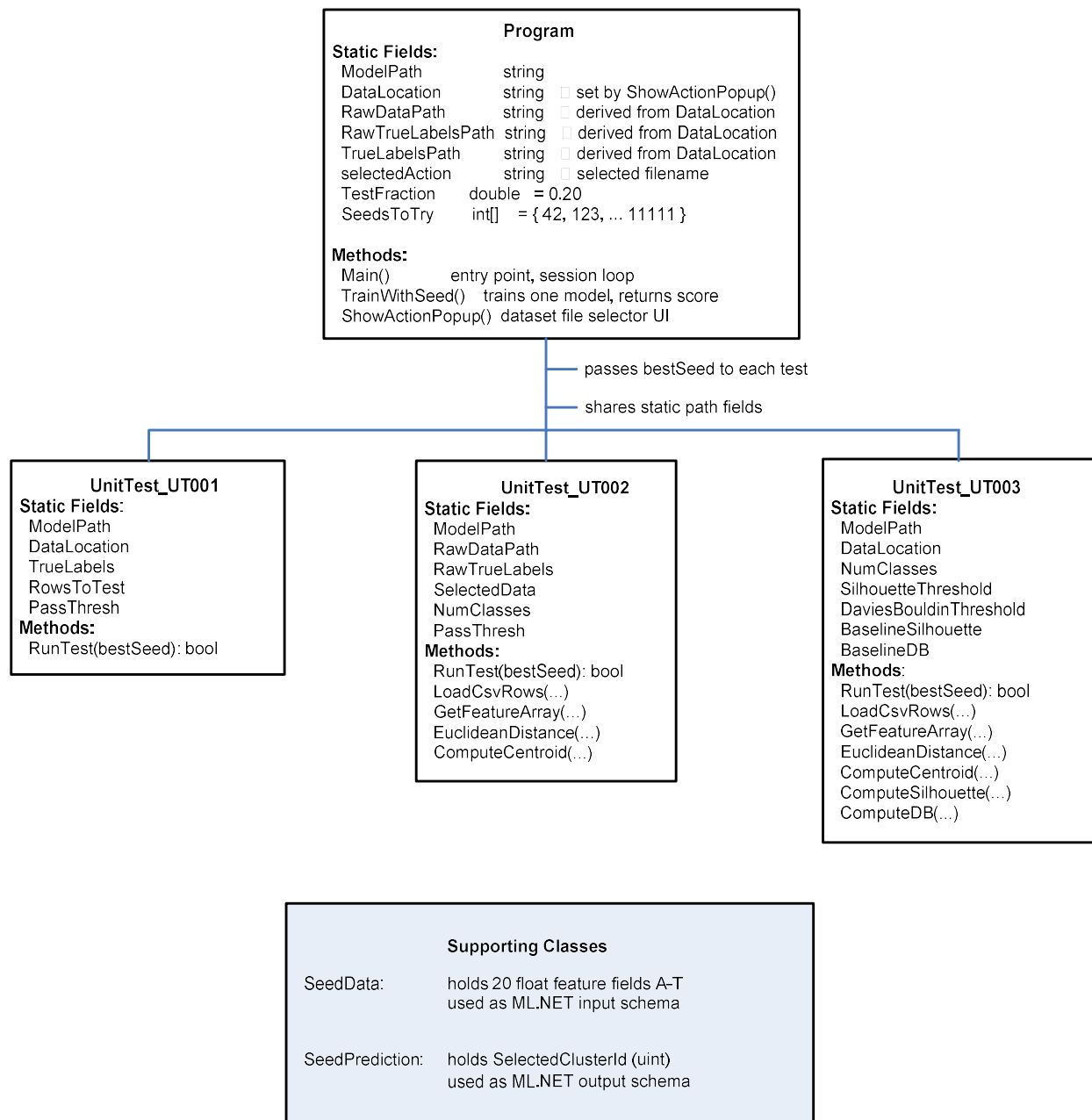
The current unit test suite is designed specifically for three-class clustering problems. Extending it to handle four, five, or more classes would broaden its applicability considerably. The primary changes required would be to the NumClasses constant, the label mapping and confusion matrix logic in Unit Test 001, the centroid matching logic in Unit Test 002, and the cluster size distribution reporting in Unit Test 003. Threshold recalibration would be needed for each new class count configuration since both metrics are sensitive to the number of clusters.

10 References

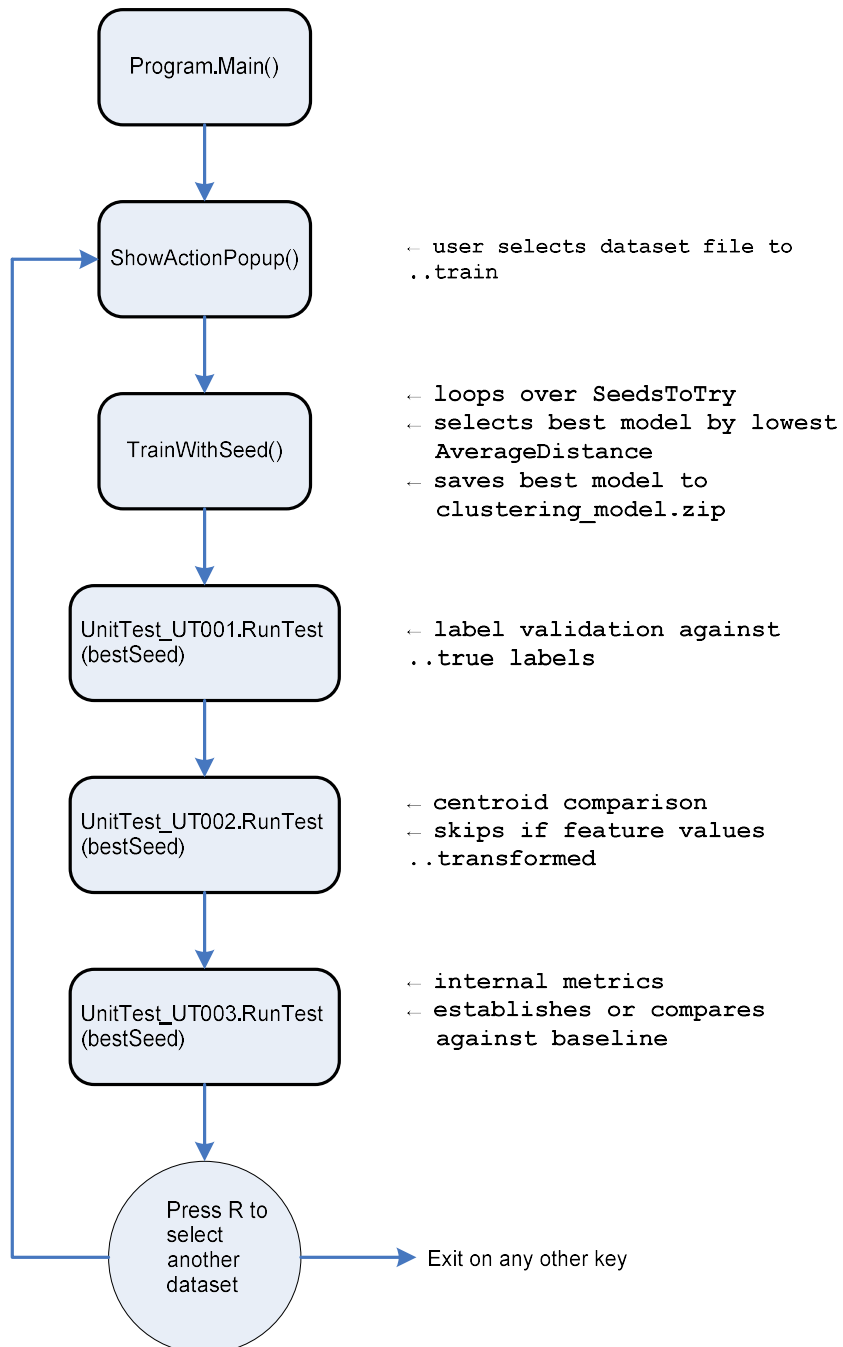
- [0.0.1] <https://learn.microsoft.com/en-us/dotnet/machine-learning/resources/tasks>
Clustering: Clustering trainer; [KMeansTrainer](#)
- [4.2.0] https://www.researchgate.net/publication/338306217_Distance_based_k-means_clustering_algorithm_for_determining_number_of_clusters_for_high_dimensional_data
January 2020 [Decision Science Letters](#) DOI:[10.5267/j.dsl.2019.8.002](#)
Mohamed Cassim Alibuhtto, South Eastern University of Sri Lanka
Nor idayu Mahat, Northern University of Malaysia
- [4.2.1] [k-means++: The Advantages of Careful Seeding](#)
<https://theory.stanford.edu/~sergei/papers/kMeansPP-soda.pdf> David Arthur, Sergei Vassilvitskii
- [7.3.0] [Rousseeuw, Peter J. \(1987\). "Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis". *Computational and Applied Mathematics*. **20**: 53–65. doi:10.1016/0377-0427\(87\)90125-7](#)
- [7.3.1] https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html
© Copyright 2007 - 2026, scikit-learn developers (BSD License).
- [7.4.0] Davies, D. L., & Bouldin, D. W. (1979). "A Cluster Separation Measure". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2), 224-227.
- [7.4.1] <https://scikit-learn.org/stable/modules/clustering.html#davies-bouldin-index>
2.3.11.7. Davies-Bouldin Index; see mathematical formulation
© Copyright 2007 - 2026, scikit-learn developers (BSD License).
- [8.1.0] https://scikit-learn.org/stable/datasets/sample_generators.html#make_classification
© Copyright 2007 - 2026, scikit-learn developers (BSD License).
- [8.4.1] <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0310839>
The impact of neglecting feature scaling in k-means clustering
Published: December 6, 2024
<https://doi.org/10.1371/journal.pone.0310839>; Chantha Wongoutong

Appendix A. Software Architecture Overview

The following diagram illustrates the class structure of the SeedClustering application. The Program class serves as the entry point and session controller, managing dataset file selection, multi-seed KMeans training, and sequential invocation of the three unit test classes. The execution flow is strictly sequential within each session run with training completing before any unit test begins, and the unit tests always executing in the order UT-001, UT-002, UT-003.



Execution Flow Within a Session



Appendix A-1. Format for input_files.txt file contents

ha-cs3_4.csv
ha-cs3_4_945f1.csv
ha-cs3_4_cacb5.csv
ha-cs3_4_36618.csv

Appendix B: Unit Test Pass/Fail Criteria Reference Table

All threshold values are defined as static fields at the top of their respective unit test classes and can be adjusted to suit different datasets or application requirements. The values shown below are those used in the experimental runs documented in Appendix C.

Unit Test 001 — Label Validation Against Ground Truth

Criterion	Threshold	Notes
Overall accuracy	$\geq 70\%$	Correct matched rows divided by total test rows
Label mapping	Unambiguous	Each true label must map to a different predicted cluster
Both criteria	Must both pass	Either failure produces a FAIL result

Unit Test 002 — Centroid Comparison

Criterion	Threshold	Notes
Centroid mapping	Unambiguous	Each true class must map to a different predicted cluster
Class distribution difference	$\leq 15\%$ per class	Absolute percentage difference between true and predicted proportions
Feature transformation check	First row distance ≤ 0.001	Test skips automatically if threshold exceeded
Both criteria	Must both pass	Either failure produces a FAIL result

Unit Test 003 — Internal Cluster Quality Metrics

Criterion	Threshold	Notes
Silhouette Coefficient	≥ 0.30	Calibrated for 20-dimensional feature space
Davies-Bouldin Index	≤ 1.50	Corresponds to Good-to-Fair boundary
Both criteria	Must both pass	Either failure produces a FAIL result

Degradation warnings (comparison runs only, do not affect PASS/FAIL):

Metric	Warning Threshold Notes	
Silhouette drop from baseline	$> 15\%$	Advisory warning only
Davies-Bouldin increase from baseline	$> 20\%$	Advisory warning only

Silhouette Coefficient Interpretation Guide

Score Range Interpretation

≥ 0.70	Strong — clusters well separated
≥ 0.50	Reasonable — clusters mostly separated
≥ 0.35	Acceptable — reasonable for high-dimensional data
≥ 0.25	Weak — clusters somewhat overlapping
< 0.25	Poor — clusters heavily overlapping

Davies-Bouldin Index Interpretation Guide

Score Range Interpretation

≤ 0.50	Excellent — very compact and well-separated clusters
≤ 1.00	Good — acceptable cluster separation
≤ 2.00	Fair — clusters somewhat overlapping
> 2.00	Poor — clusters heavily overlapping or scattered

This appendix serves as a quick reference for interpreting unit test output without needing to refer back to the detailed descriptions in Sections 5 through 7.

Appendix C-1: Baseline Run: ha-cs3_4.csv

Sanitization applied: None (raw unmodified dataset) Rows: 300 | Best seed: 5 | Best AverageDistance score: 1.5296

Multi-Seed Training

Best seed value: 5 out of 19 seeds tried

Best score: 1.5296

UT-001 — Label Validation

Label mapping (True -> Predicted):

True Label 0 -> Predicted Cluster 3 (matched 14 times)

True Label 1 -> Predicted Cluster 2 (matched 24 times)

True Label 2 -> Predicted Cluster 1 (matched 22 times)

Label mapping is unambiguous. OK

Row | True Label | Predicted | Mapped | Match

1 | 2 | 1 | 1 | YES

2 | 0 | 3 | 3 | YES

3 | 0 | 3 | 3 | YES

4 | 2 | 1 | 1 | YES

5 | 1 | 2 | 2 | YES

6 | 1 | 2 | 2 | YES

55 | 0 | 3 | 3 | YES

56 | 2 | 1 | 1 | YES

57	2	1	1	YES
58	1	2	2	YES
59	2	1	1	YES
60	0	3	3	YES

Correct: 60 / 60

Accuracy: 100.0%

Threshold:70.0%

Label Mapping Used: {0->3, 1->2, 2->1}

UT-001 RESULT: PASS

UT-002 — Centroid Comparison

Feature transformation check:

First row distance (raw vs selected): 0.000000

Feature values are unchanged. OK

True class centroids (from raw dataset):

True class 0: 100 rows centroid computed. OK

True class 1: 100 rows centroid computed. OK

True class 2: 100 rows centroid computed. OK

Predicted cluster centroids (from selected dataset):

Predicted cluster 1: 100 rows centroid computed. OK

Predicted cluster 2: 100 rows centroid computed. OK

Predicted cluster 3: 100 rows centroid computed. OK

Centroid matching:

True Class	Best Match	Distance	Threshold	Status

0	3	0.0000	2.0000	OK
1	2	0.0000	2.0000	OK
2	1	0.0000	2.0000	OK

Centroid mapping is unambiguous. OK

Average centroid distance: 0.0000

Class distribution comparison:

Class	True N	True%	Pred N	Pred%	Diff%	Status

0	100	33.3%	100	33.3%	0.0%	OK
1	100	33.3%	100	33.3%	0.0%	OK
2	100	33.3%	100	33.3%	0.0%	OK

UT-002 RESULT: PASS

Centroid mapping: unambiguous

Average centroid distance: 0.0000

Class distribution: within tolerance

UT-003 — Internal Metrics

Mode: BASELINE RUN

Cluster size distribution:

Cluster 1: 100 rows (33.3%)

Cluster 2: 100 rows (33.3%)

Cluster 3: 100 rows (33.3%)

Inter-centroid distances (in 20D space):

Cluster Pair	Distance
--------------	----------

1 <-> 2	18.6863
---------	---------

1 <-> 3	17.8838
---------	---------

2 <-> 3	17.7570
---------	---------

Per-cluster intra-cluster scatter:

Cluster	Avg Dist to Centroid
---------	----------------------

1	8.2647
---	--------

2	8.0443
---	--------

3	8.7920
---	--------

Davies-Bouldin Index: 0.9519 (Good)

Pass threshold: <= 1.5000

Davies-Bouldin status: OK

Per-cluster Silhouette scores:

Cluster	Avg Silhouette	Interpretation
---------	----------------	----------------

1	0.4359	Acceptable
2	0.4526	Acceptable
3	0.3908	Acceptable

Overall Silhouette Score: 0.4264 (Acceptable)

Pass threshold: ≥ 0.3000

Silhouette status: OK

Metric summary:

Metric	Score	Threshold	Direction	Interpretation	Status
Silhouette Coefficient	0.4264	≥ 0.3000	higher=better	Acceptable	OK
Davies-Bouldin Index	0.9519	≤ 1.5000	lower=better	Good	OK

Step 8: Baseline established:

Dataset: ha-cs3_4.csv

Silhouette baseline: 0.4264

Davies-Bouldin baseline:0.9519

UT-003 RESULT: PASS

Silhouette Coefficient: 0.4264 ≥ 0.3000 (Acceptable)

Davies-Bouldin Index: 0.9519 ≤ 1.5000 (Good)

Baseline established. Run again with a conditioned dataset to compare.

Appendix C-2: Conditioned Variant 1: ha-cs3_4_945f1.csv

Sanitization applied: Min-Max Normalization (Section 2) Rows: 300 (no rows dropped) | Best seed: 7 | Best AverageDistance score: 0.4376

Multi-Seed Training

Best seed value: 7 out of 19 seeds tried

Best score: 0.4376

UT-001 — Label Validation

Label mapping (True -> Predicted):

True Label 0 -> Predicted Cluster 2 (matched 14 times)

True Label 1 -> Predicted Cluster 1 (matched 24 times)

True Label 2 -> Predicted Cluster 3 (matched 22 times)

Label mapping is unambiguous. OK

Row | True Label | Predicted | Mapped | Match

1 | 2 | 3 | 3 | YES

2 | 0 | 2 | 2 | YES

3 | 0 | 2 | 2 | YES

4 | 2 | 3 | 3 | YES

5 | 1 | 1 | 1 | YES

6 | 1 | 1 | 1 | YES

55 | 0 | 2 | 2 | YES

56 | 2 | 3 | 3 | YES

57	2	3	3	YES
58	1	1	1	YES
59	2	3	3	YES
60	0	2	2	YES

Correct: 60 / 60

Accuracy: 100.0%

Threshold:70.0%

Label Mapping Used: {0->2, 1->1, 2->3}

UT-001 RESULT: PASS

UT-002 — Centroid Comparison

Feature transformation check:

First row distance (raw vs selected): 14.560499

UT-002 SKIPPED: Feature values have been transformed.

First row Euclidean distance from raw: 14.560499

This indicates a Section 2 scaling or Section 3
dimensionality reduction operation was applied.

Only Section 1 outlier removal (row dropping)
is compatible with this test.

UT-003 — Internal Metrics

Mode: COMPARISON RUN (baseline from: ha-cs3_4.csv)

Cluster size distribution:

Cluster 1: 100 rows (33.3%)

Cluster 2: 100 rows (33.3%)

Cluster 3: 100 rows (33.3%)

Inter-centroid distances (in 20D space):

Cluster Pair	Distance
--------------	----------

1 <-> 2	1.0022
---------	--------

1 <-> 3	1.0818
---------	--------

2 <-> 3	1.0683
---------	--------

Per-cluster intra-cluster scatter:

Cluster	Avg Dist to Centroid
---------	----------------------

1	0.6211
---	--------

2	0.6433
---	--------

3	0.6269
---	--------

Davies-Bouldin Index: 1.2374 (Fair)

Pass threshold: <= 1.5000

Davies-Bouldin status: OK

Per-cluster Silhouette scores:

Cluster	Avg Silhouette	Interpretation
---------	----------------	----------------

1	0.3336	Weak
2	0.3069	Weak
3	0.3404	Weak

Overall Silhouette Score: 0.3270 (Weak)

Pass threshold: ≥ 0.3000

Silhouette status: OK

Metric summary:

Metric	Score	Threshold	Direction	Interpretation	Status
Silhouette Coefficient	0.3270	≥ 0.3000	higher=better	Weak	OK
Davies-Bouldin Index	1.2374	≤ 1.5000	lower=better	Fair	OK

Baseline comparison:

Metric	Baseline	Current	Change	Status

Silhouette Coefficient	0.4264	0.3270	-23.3%	OK
Davies-Bouldin Index	0.9519	1.2374	30.0%	WARN

WARNING: Davies-Bouldin increased 30.0% from baseline (limit: 20%).

Consider a less aggressive conditioning variant.

UT-003 RESULT: PASS

Silhouette Coefficient: 0.3270 ≥ 0.3000 (Weak)

Davies-Bouldin Index: 1.2374 ≤ 1.5000 (Fair)

Appendix C-3: Conditioned Variant 2: ha-cs3_4_cacb5.csv

Sanitization applied: Z-Score Outlier Removal (Section 1) Rows: 265 (35 rows dropped) | Best seed: 456 | Best AverageDistance score: 1.8194

Multi-Seed Training

Best seed value: 456 out of 19 seeds tried

Best score: 1.8194

UT-001 — Label Validation

Label mapping (True -> Predicted):

True Label 0 -> Predicted Cluster 1 (matched 14 times)

True Label 1 -> Predicted Cluster 2 (matched 24 times)

True Label 2 -> Predicted Cluster 3 (matched 22 times)

Label mapping is unambiguous. OK

Row | True Label | Predicted | Mapped | Match

1 | 2 | 3 | 3 | YES

2 | 0 | 1 | 1 | YES

3 | 0 | 1 | 1 | YES

4 | 2 | 3 | 3 | YES

5 | 1 | 2 | 2 | YES

6 | 1 | 2 | 2 | YES

55 | 0 | 1 | 1 | YES

56 | 2 | 3 | 3 | YES

57	2	3	3	YES
58	1	2	2	YES
59	2	3	3	YES
60	0	1	1	YES

Correct: 60 / 60

Accuracy: 100.0%

Threshold:70.0%

Label Mapping Used: {0->1, 1->2, 2->3}

UT-001 RESULT: PASS

UT-002 — Centroid Comparison

Feature transformation check:

First row distance (raw vs selected): 0.000000

Feature values are unchanged. OK

35 rows were dropped during sanitization.

True class centroids (from raw dataset):

True class 0: 100 rows centroid computed. OK

True class 1: 100 rows centroid computed. OK

True class 2: 100 rows centroid computed. OK

Predicted cluster centroids (from selected dataset):

Predicted cluster 1: 91 rows centroid computed. OK

Predicted cluster 2: 83 rows centroid computed. OK

Predicted cluster 3: 91 rows centroid computed. OK

Centroid matching:

True Class	Best Match	Distance	Threshold	Status

0	1	0.5177	2.0000	OK
1	2	0.6528	2.0000	OK
2	3	0.3614	2.0000	OK

Centroid mapping is unambiguous. OK

Average centroid distance: 0.5106

Class distribution comparison:

Class	True N	True%	Pred N	Pred%	Diff%	Status

0	100	33.3%	91	34.3%	1.0%	OK
1	100	33.3%	83	31.3%	2.0%	OK
2	100	33.3%	91	34.3%	1.0%	OK

UT-002 RESULT: PASS

Centroid mapping: unambiguous

Average centroid distance: 0.5106

Class distribution: within tolerance

UT-003 — Internal Metrics

Mode: COMPARISON RUN (baseline from: ha-cs3_4.csv)

Cluster size distribution:

Cluster 1: 91 rows (34.3%)

Cluster 2: 83 rows (31.3%)

Cluster 3: 91 rows (34.3%)

Inter-centroid distances (in 20D space):

Cluster Pair	Distance
--------------	----------

1 <-> 2	17.3739
---------	---------

1 <-> 3	17.6906
---------	---------

2 <-> 3	18.3858
---------	---------

Per-cluster intra-cluster scatter:

Cluster	Avg Dist to Centroid
---------	----------------------

1	8.6014
---	--------

2	7.6376
---	--------

3	8.1227
---	--------

Davies-Bouldin Index: 0.9418 (Good)

Pass threshold: <= 1.5000

Davies-Bouldin status: OK

Per-cluster Silhouette scores:

Cluster	Avg Silhouette	Interpretation
---------	----------------	----------------

1	0.3913	Acceptable
2	0.4683	Acceptable
3	0.4356	Acceptable

Overall Silhouette Score: 0.4306 (Acceptable)

Pass threshold: ≥ 0.3000

Silhouette status: OK

Metric summary:

Metric	Score	Threshold	Direction	Interpretation	Status
Silhouette Coefficient	0.4306	≥ 0.3000	higher=better	Acceptable	OK
Davies-Bouldin Index	0.9418	≤ 1.5000	lower=better	Good	OK

Baseline comparison:

Metric	Baseline	Current	Change	Status

Silhouette Coefficient	0.4264	0.4306	1.0%	OK
Davies-Bouldin Index	0.9519	0.9418	-1.1%	OK

Conditioning degradation is within acceptable limits. OK

UT-003 RESULT: PASS

Silhouette Coefficient: 0.4306 ≥ 0.3000 (Acceptable)

Davies-Bouldin Index: 0.9418 ≤ 1.5000 (Good)

Appendix C-4: Conditioned Variant 3: ha-cs3_4_36618.csv

Sanitization applied: Standardization with Outlier Removal (Sections 1 and 2) Rows: 265 (35 rows dropped) | Best seed: 11111 | Best AverageDistance score: 0.5378

Multi-Seed Training

Best seed value: 11111 out of 19 seeds tried

Best score: 0.5378

UT-001 — Label Validation

Label mapping (True -> Predicted):

True Label 0 -> Predicted Cluster 3 (matched 14 times)

True Label 1 -> Predicted Cluster 2 (matched 24 times)

True Label 2 -> Predicted Cluster 1 (matched 22 times)

Label mapping is unambiguous. OK

Row | True Label | Predicted | Mapped | Match

1 | 2 | 1 | 1 | YES

2 | 0 | 3 | 3 | YES

3 | 0 | 3 | 3 | YES

4 | 2 | 1 | 1 | YES

5 | 1 | 2 | 2 | YES

6 | 1 | 2 | 2 | YES

55 | 0 | 3 | 3 | YES

56 | 2 | 1 | 1 | YES

57	2	1	1	YES
58	1	2	2	YES
59	2	1	1	YES
60	0	3	3	YES

Correct: 60 / 60

Accuracy: 100.0%

Threshold:70.0%

Label Mapping Used: {0->3, 1->2, 2->1}

UT-001 RESULT: PASS

UT-002 — Centroid Comparison

Feature transformation check:

First row distance (raw vs selected): 14.538490

UT-002 SKIPPED: Feature values have been transformed.

First row Euclidean distance from raw: 14.538490

This indicates a Section 2 scaling or Section 3
dimensionality reduction operation was applied.

Only Section 1 outlier removal (row dropping)
is compatible with this test.

UT-003 — Internal Metrics

Mode: COMPARISON RUN (baseline from: ha-cs3_4.csv)

Cluster size distribution:

Cluster 1: 91 rows (34.3%)

Cluster 2: 83 rows (31.3%)

Cluster 3: 91 rows (34.3%)

Inter-centroid distances (in 20D space):

Cluster Pair	Distance
--------------	----------

1 <-> 2	1.1636
---------	--------

1 <-> 3	1.1414
---------	--------

2 <-> 3	1.0940
---------	--------

Per-cluster intra-cluster scatter:

Cluster	Avg Dist to Centroid
---------	----------------------

1	0.7077
---	--------

2	0.6756
---	--------

3	0.7201
---	--------

Davies-Bouldin Index: 1.2675 (Fair)

Pass threshold: <= 1.5000

Davies-Bouldin status: OK

Per-cluster Silhouette scores:

Cluster	Avg Silhouette	Interpretation
---------	----------------	----------------

1	0.3167	Weak
2	0.3403	Weak
3	0.2907	Weak

Overall Silhouette Score: 0.3152 (Weak)

Pass threshold: ≥ 0.3000

Silhouette status: OK

Metric summary:

Metric	Score	Threshold	Direction	Interpretation	Status
Silhouette Coefficient	0.3152	≥ 0.3000	higher=better	Weak	OK
Davies-Bouldin Index	1.2675	≤ 1.5000	lower=better	Fair	OK

Baseline comparison:

Metric	Baseline	Current	Change	Status

Silhouette Coefficient	0.4264	0.3152	-26.1%	OK
Davies-Bouldin Index	0.9519	1.2675	33.2%	WARN

WARNING: Davies-Bouldin increased 33.2% from baseline (limit: 20%).

Consider a less aggressive conditioning variant.

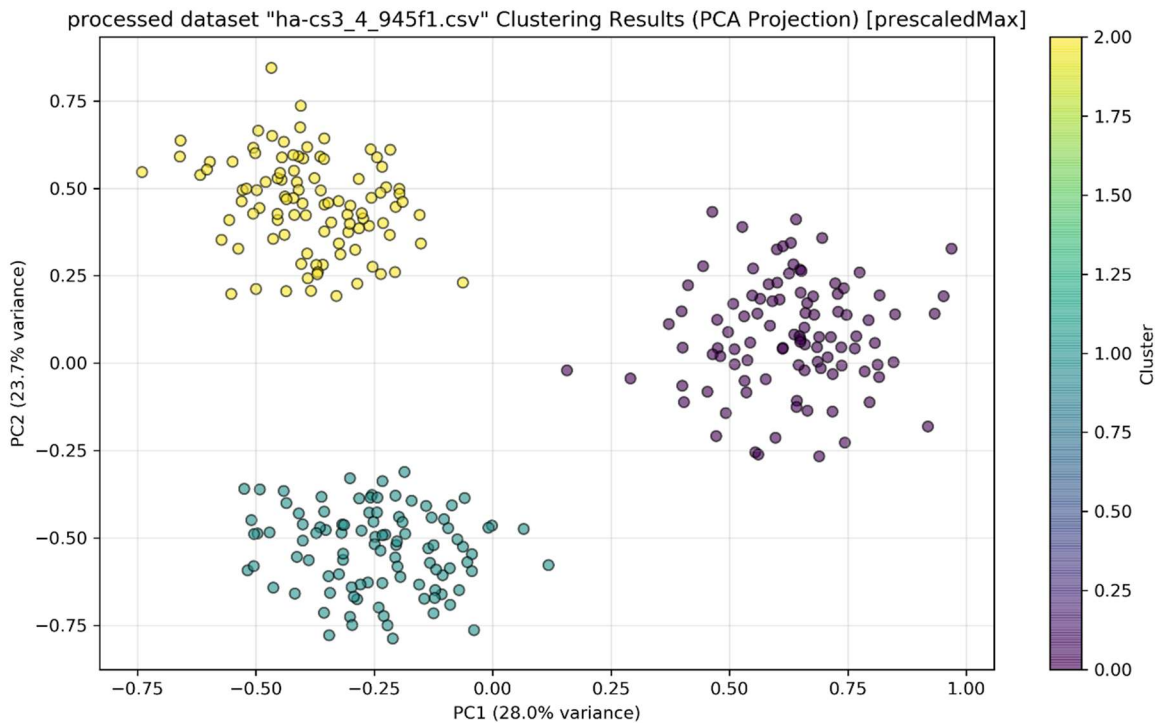
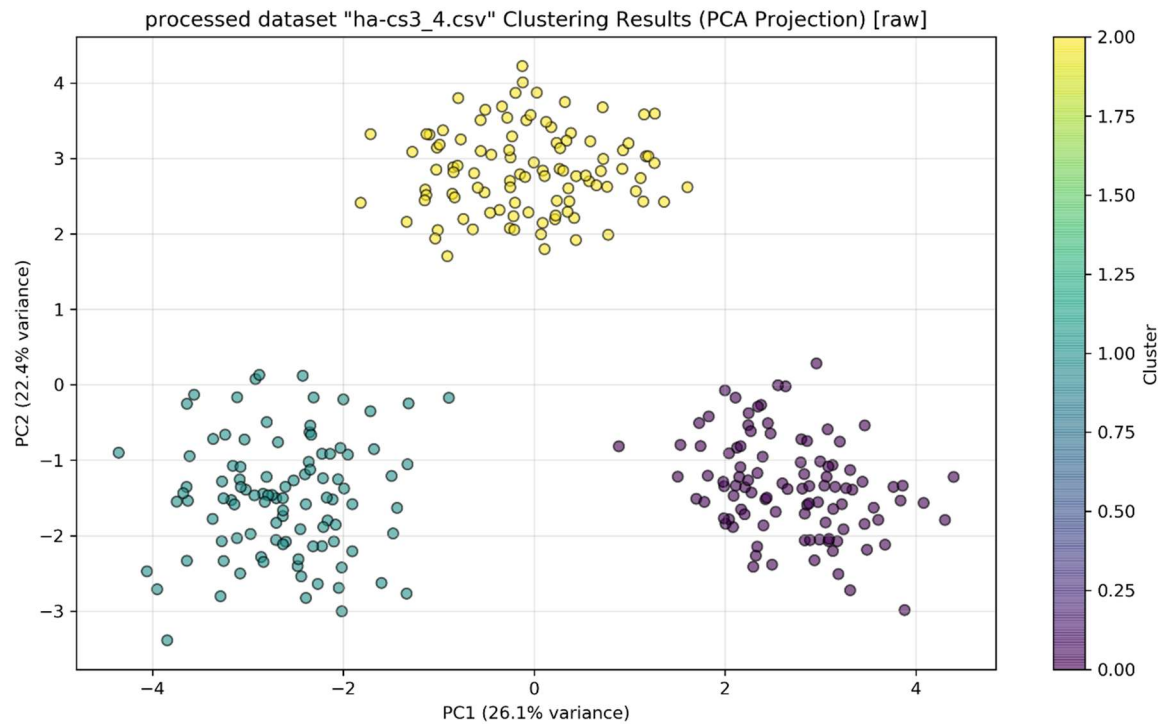
UT-003 RESULT: PASS

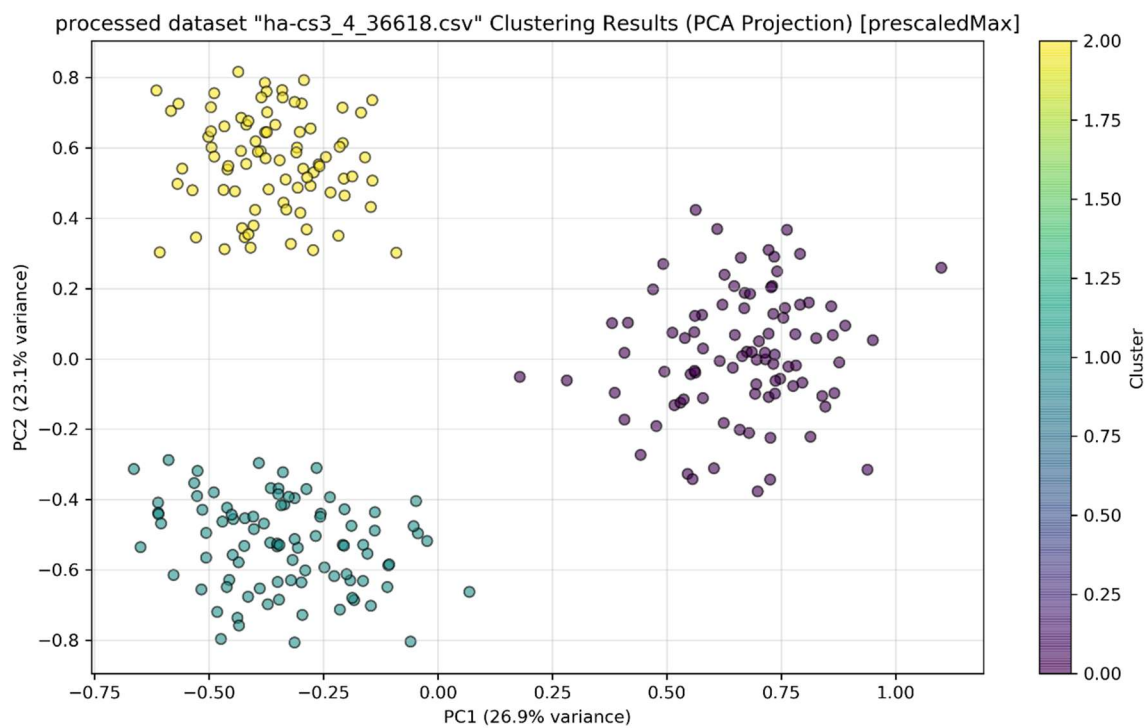
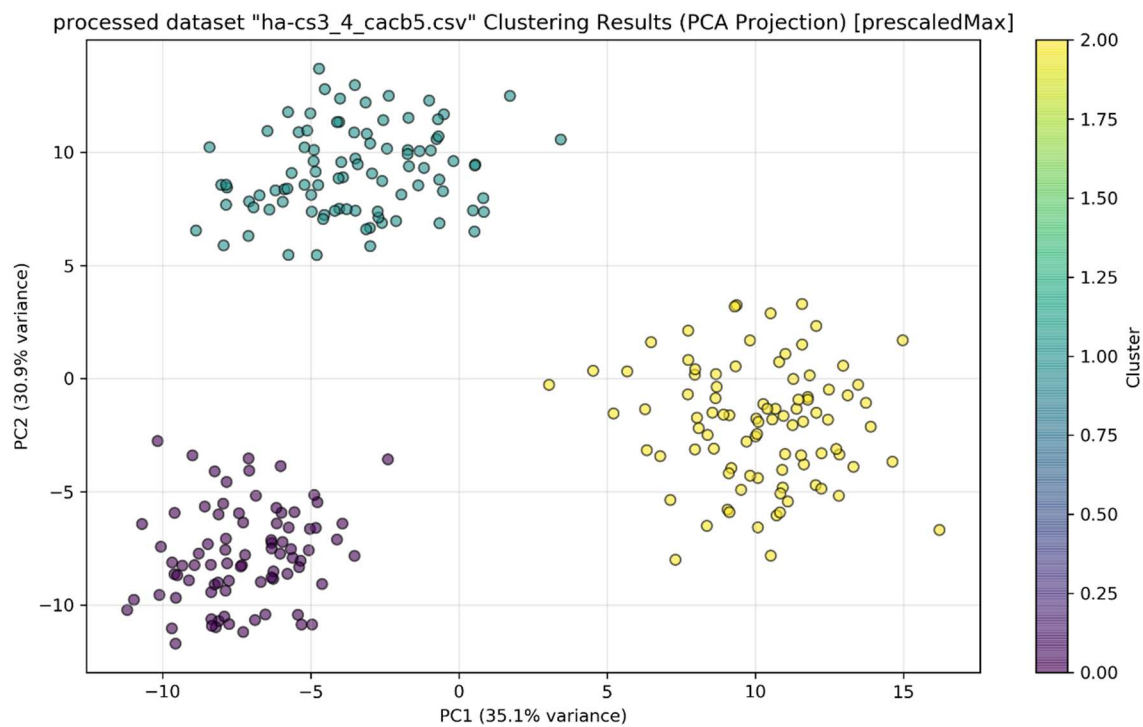
Silhouette Coefficient: 0.3152 ≥ 0.3000 (Weak)

Davies-Bouldin Index: 1.2675 ≤ 1.5000 (Fair)

Appendix D: PCA Plots

Note: Cluster color assignment is random, cluster position is per identified class.





Appendix E: sklearn make_classification Parameter Reference and ha-cs3_4.py script

The synthetic datasets used in this report were generated using the make_classification function from the Python scikit-learn library. The table below lists the key parameters used and their values.

Parameter	Value Used	Description
n_samples	300	Total number of data rows generated
n_features	20	Total number of feature columns per row
n_informative	20	Number of features carrying class information
n_redundant	0	Number of redundant features (linear combinations of informative features)
n_classes	3	Number of distinct classes to generate
n_clusters_per_class	1	Number of clusters per class
class_sep	3.0	Controls separation between class centers — higher values produce more widely separated clusters
random_state	fixed integer	Controls reproducibility of the generated dataset

The companion true labels file was produced by saving the integer class label array returned alongside the feature matrix by make_classification, with one label per row aligned row-for-row with the main dataset file.

For full parameter documentation see the official scikit-learn reference at: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_classification.html

ha-cs3_4.py script:

```
# ha-cs3_4.py 'header_only'
# Modified 5/4/26 JB: for use with .Net8; csX_X in name is class separation
from sklearn.datasets import make_classification
import numpy as np
import pandas as pd
import string
import os

cfile_name = "ha-cs3_4.csv"
cfile_bare = ""
# — Output format control —————
# Set OUTPUT_MODE to one of:
# 'header_only'    - row 1 header, no index column
# 'index_only'     - row 1 no header, index column
# 'header_and_index' - row 1 header + index column
# 'data_only'      - no header, no index
OUTPUT_MODE = 'header_only'
# —————

# Generate classification data with controlled complexity
X, y = make_classification(
    n_samples=300,
    n_features=20,
    n_classes=3,
    n_informative=15,
    n_redundant=0,
    n_repeated=0,
    n_clusters_per_class=1,
    class_sep=3.0,
    flip_y=0.00,
    random_state=1
)
n_features = 20
feature_names = list(string.ascii_uppercase[:n_features])
df = pd.DataFrame(X, columns=feature_names)
```

```

# Apply output format based on OUTPUT_MODE
if OUTPUT_MODE == 'header_only':
    df.to_csv(cfile_name, index=False, header=True)
    # Save true labels separately (for evaluating UI performance later)
    #np.savetxt(f"{cfile_name}", y, delimiter=',', fmt='%d')
    cfile_bare, extension = os.path.splitext(cfile_name)
    np.savetxt(f"{cfile_bare}_true_labels.csv", y, delimiter=',', fmt='%d')
else:
    raise ValueError(f"Unknown OUTPUT_MODE: '{OUTPUT_MODE}'. Choose 'header_only',
'header_and_index', or 'data_only'.")

cfile_bare, extension = os.path.splitext(cfile_name)
print(f"bare = {cfile_bare}")
print(f"Generated {X.shape[0]} samples with {X.shape[1]} features")
print(f"Data saved to {cfile_bare}.csv")
print(f"True labels saved to {cfile_bare}_true_labels.csv (for validation)")

```