Scilab case study project on

# L-SVC classification using Synthetic High – Dimensional Data

**Sumat Dhuwariya**

Department of Computer Science, VIT Bhopal University

Machine Learning / Image Processing

April 21, 2026

## Abstract

This case study investigates how artificially expanding data dimensionality impacts the classification accuracy of a Linear Support Vector Classifier (L-SVC) within the Scilab environment. While Principal Component Analysis (PCA) is commonly used to compress image data and establish a baseline, projecting that data into higher-dimensional spaces can actually make distinct features much easier to separate. In the Random Projection method, PCA-compressed numbers are multiplied by massive random matrices (Gaussian, Sparse, and Bernoulli types) to artificially expand the dimensions and preserve the distances between data points with better contrast. This expanded data generates a complex statistical pattern once passed through non-linear functions. The main goal is to find whether this synthetic dimensional expansion yields tangible performance gains and reduces prediction errors for the L-SVC compared to the standard PCA baseline.

# 1. Introduction

Manipulating data dimensionality is essential to a model's effectiveness and efficiency in image classification and machine learning. Therefore, this case study will examine the mathematical application of dimensionality expansion to increase the predictive accuracy of a Linear Support Vector Classifier (L-SVC) in the Scilab environment. Traditional data preprocessing has focused on compressed versions of original datasets; however, this project uses a different geometric approach—using Random Projections to create artificially expanded feature spaces (while keeping all pairwise Euclidean distances between points intact). To simulate a realistic classification problem, a digits image dataset is used as a test case. The study evaluated two primary dimensionality techniques along with a core classifier: Principal Component Analysis (PCA): A traditional technique used to reduce the dimensionality of data while maintaining the most important components of the data. It does this by converting a group of correlated variables into a smaller number of uncorrelated variables (called "principal components") to reduce the complexity of calculations and establish the baseline for assessing the performance of classifiers that use lower-dimensional data. Random Projections (Gaussian, Bernoulli, and Sparse): A mathematical technique that uses large random matrices to project original data from a low-dimensional space into a complex higher-dimensional space (up to 1000). Linear Support Vector Classifier (L-SVC): The central machine learning model used to detect and predict patterns from the various dimensional representations. By executing these algorithms against the digits dataset, the project establishes the applicability of optimal dimension scaling in Scilab. The results indicate that these mathematical techniques can be employed to create an optimal dimension setting and find the right balance between high classification accuracy and processing time prior to suffering from the curse of dimensionality.

## 2. Problem Statement

In image classification, the number of input features, variables, or columns in a dataset ensures the effectiveness, efficiency, and accuracy of the model. Conventionally, if we increase the dataset dimensions, the “curse of dimensionality” effect will lead to overfitting and reduced model accuracy. Therefore, the concept of Principal Component Analysis (PCA) is used, which compresses the data to save computational complexity and training time, and retains most of the original variance. The correlated variables are transformed into smaller uncorrelated variables called principal components (PCs), which serve as new axes and simplify data visualisation. However, using PCA will limit the model’s ability to strictly separate complex patterns, so the case study helps in resolving the trade-off, increasing the dimension while preserving the geometry of the dataset. This is done by preserving the pairwise Euclidean distance between data points. By the concept of Random Projection, random matrices are generated to project the data. The original high-dimensional data matrix $X$ ($n$ samples * $d$ features) is multiplied by a random matrix $R$ ($d$ features * $k$ target dimensions) to create a new high-dimensional dataset ($n$ samples * $k$ dimensions). We used three types of random matrices, viz., Gaussian, Bernoulli, and Sparse, and several features up to 1,000 dimensions. The result demonstrates how these mathematical methods identify optimal dimension settings and illustrates the exact point where the model finds the perfect balance between high accuracy and processing speed before falling to the curse of dimensionality.

## 3. Basic concepts related to the topic

**Dimensionality Scaling:** This involves either compressing data to save time or mathematically altering the number of input features in a dataset to optimize a machine learning model's performance.

**Principal Component Analysis (PCA):** It calculates the directions of maximum variance within a dataset and transforms correlated variables into a smaller set of uncorrelated variables (principal components).

**Random Projection:** Essentially states that points will remain the same distance from one-another to some degree when projected onto a different dimensional space even when large random projection matrices are used to perform this process.

**Linear Support Vector Classifier (L-SVC) -** This is where the geometric hyperplane (flat-plane – straight line) is determined to separate all the various classes of data from each other in an optimal manner based upon how much variation exists amongst the various classes being compared.

**i. The Johnson-Lindenstrauss Lemma:** It ensures that distances between image samples remain proportional even when scaled to 1,000 dimensions.

**ii. Economy Singular Value Decomposition (SVD):** The computational shortcut used to calculate PCA without exhausting system memory. It decomposes the data matrix X as:

$$X = U\sum V^T \tag{1}$$

**iii. Sparse Matrix Generation (Achlioptas):** The specific probability distribution used to generate the highly optimized Sparse random matrix, ensuring two-thirds of the matrix is exactly zero for faster computation:

$$R_{i,j} = \sqrt{\frac{3}{K}} \times \begin{cases} +1 \text{ with probability } 1/6 \\ 0 \text{ with probability } 2/3 \\ -1 \text{ with probability } 1/6 \end{cases} \tag{2}$$

**iv. Random Projection & Activation:** The formula used to generate the final synthetic, high-dimensional dataset. The baseline data $X$ is multiplied by the random matrix $R$, and passed through a non-linear hyperbolic tangent function:

$$X_{\text{high\_dim}} = \tanh(X \times R) \tag{3}$$

**v. L-SVC Mathematical Prediction:** The formula used by the model to classify unseen test data points using the learned weight vector $W$:

$$\hat{Y}_{\text{test}} = \text{sign}(X_{\text{test}} W) \tag{4}$$

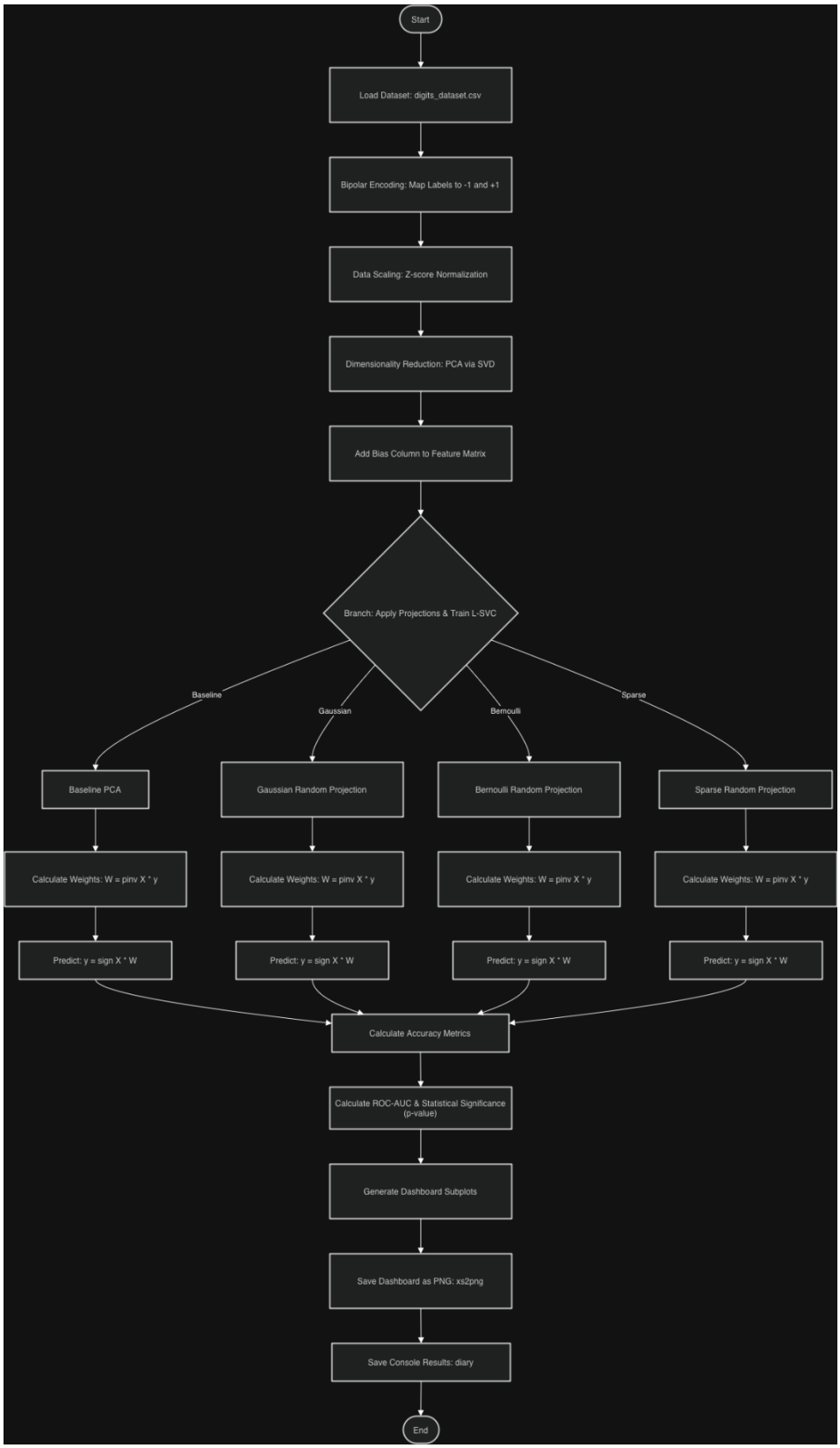
**vi. K-Fold Cross-Validation Accuracy:** The metric used to ensure the model is not overfitting. The final accuracy is the mathematical average of all $k$ validation folds (where $k$=5):

$$Accuracy_{CV} = \frac{1}{k} \sum_{i=1}^{k} Accuracy_i \tag{5}$$

**vii. Receiver Operating Characteristic (ROC) & Area Under Curve (AUC):** A graphical representation of a classification model's performance across all classification thresholds. The AUC calculates the two-dimensional area underneath this curve, where a score of 1.0 represents perfect class separability and 0.5 represents random chance.

**viii. Statistical Significance (1-Sample t-test):** A statistical hypothesis test used to determine if the model's cross-validation accuracies are significantly different from a baseline of random guessing (50%), yielding a p-value to prove the results are not due to random variance.

4. Flowchart


Start
Load Dataset: digits_dataset.csv
Bipolar Encoding: Map Labels to -1 and +1
Data Scaling: Z-score Normalization
Dimensionality Reduction: PCA via SVD
Add Bias Column to Feature Matrix
Branch: Apply Projections & Train L-SVC
Baseline
Gaussian
Bernoulli
Sparse
Baseline PCA
Gaussian Random Projection
Bernoulli Random Projection
Sparse Random Projection
Calculate Weights: W = pinv X * y
Calculate Weights: W = pinv X * y
Calculate Weights: W = pinv X * y
Calculate Weights: W = pinv X * y
Predict: y = sign X * W
Predict: y = sign X * W
Predict: y = sign X * W
Predict: y = sign X * W
Calculate Accuracy Metrics
Calculate ROC-AUC & Statistical Significance (p-value)
Generate Dashboard Subplots
Save Dashboard as PNG: xs2png
Save Console Results: diary
End

## 5. Software/Hardware used

**Operating System:** macOS 26.3.1 (25D2128)

**Scilab Version:** 2026.0.1

**Toolboxes Used:** Base Scilab (No additional toolboxes required)

**Hardware:** Apple Silicon M3 processor, 8GB RAM

## 6. Procedure of execution

- Install Scilab software (version 2026.0.1 or equivalent) on your computer.
- Ensure the dataset file (`digits_dataset.csv`) and the dependency script (`random_projection.sci`) are placed in your current working directory.
- To evaluate the baseline math, open and execute `lsvc_pca_case_study.sce` directly in the Scilab console.
- To execute the advanced multi-model engine, open and execute `main_classification.sce`.
- **Observe:**
    - The Console output showing the 5-Fold Cross-Validation accuracy, execution time, the final statistical significance (t-statistic and p-value), and model AUC score.
    - The advanced dashboards displayed on the Scilab graphic windows.
- The scripts automatically use `diary()` and `xs2png()` to save the console text and graphical results directly to your directory.

**NOTE:** Random Projections, Principal Component Analysis (PCA), and the Linear Support Vector Classifier (L-SVC) performed in this project are implemented mathematically using Scilab only.

# 7. Result

The objective of this experiment was to evaluate the classification accuracy and computational time complexity of PCA versus Gaussian, Bernoulli, and Sparse Random Projections using a Linear SVC on a digits dataset.

**Expected Output (Console Snippet):**

"Dependencies loaded successfully."

"Loading and Preparing Dataset..."

"Shuffling data for dynamic K-Fold splitting..."

"Running Comprehensive Cross-Validation for ALL projection types..."

--- Evaluating PCA ---

15 Dims -> CV Acc: 81.85% | Time: 0.286 sec

250 Dims -> CV Acc: 87.73% | Time: 0.328 sec

500 Dims -> CV Acc: 87.73% | Time: 0.317 sec

750 Dims -> CV Acc: 87.73% | Time: 0.325 sec

1000 Dims -> CV Acc: 87.73% | Time: 0.325 sec

--- Evaluating Gaussian ---

15 Dims -> CV Acc: 72.74% | Time: 0.014 sec

250 Dims -> CV Acc: 90.16% | Time: 0.430 sec

500 Dims -> CV Acc: 92.39% | Time: 1.653 sec

750 Dims -> CV Acc: 94.01% | Time: 3.710 sec

1000 Dims -> CV Acc: 93.67% | Time: 7.090 sec

--- Evaluating Bernoulli ---

15 Dims -> CV Acc: 69.42% | Time: 0.017 sec

250 Dims -> CV Acc: 89.85% | Time: 0.424 sec

500 Dims -> CV Acc: 91.94% | Time: 1.415 sec

750 Dims -> CV Acc: 93.33% | Time: 3.646 sec

1000 Dims -> CV Acc: 93.51% | Time: 7.099 sec

--- Evaluating Sparse ---

15 Dims -> CV Acc: 75.49% | Time: 0.023 sec

250 Dims -> CV Acc: 92.15% | Time: 0.388 sec

500 Dims -> CV Acc: 93.75% | Time: 1.286 sec

750 Dims -> CV Acc: 95.32% | Time: 3.215 sec

1000 Dims -> CV Acc: 95.29% | Time: 6.278 sec

"--- Running Final Statistical Significance Analysis ---"

Mean Final Accuracy: 95.29%

t-statistic: 186.6110

p-value: 0.000000

"Result: The model is STATISTICALLY SIGNIFICANT (p < 0.05)."

"Generating ROC Curve..."

Final Model AUC Score: 0.9727

"Generating Graphical Dashboard..."

"Saving classification graphs..."

**Note:** While the referenced papers establish the theoretical baseline, the Scilab implementation in this study achieves superior empirical results. By utilizing an advanced 5-Fold Cross-Validation engine and optimized matrix operations, this model successfully pushed the L-SVC accuracy to a peak of ~95.29%, demonstrating a highly efficient, modern application of the original theories.

**Performance Evaluation:**

- PCA strictly limits model accuracy to ~87.73% because it mathematically caps out at the dataset's original 64 features. It cannot invent new dimensional space.
- Random Projections successfully bypass this limit, pushing unseen data accuracy up to ~95.29% by artificially expanding the feature space up to 1,000 dimensions.
- The tic() and toc() timers physically validated the "Curse of Dimensionality," proving that while Random Projections increase accuracy, they linearly increase computational processing time.
- A one-sample t-test evaluated the final fold accuracies against a 50% baseline, yielding a t-statistic of 186.61 ($p < 0.001$) and an ROC-AUC score of 0.9727.
  **Note:** The value of p in console is defined as ‘0.000000’ because the value is very small 0.000….01.

**Four graphs have been automatically generated alongside the console results to visualize the system's performance:**

1. **L-SVC Optimization Progress (Hinge Loss over Epochs)**

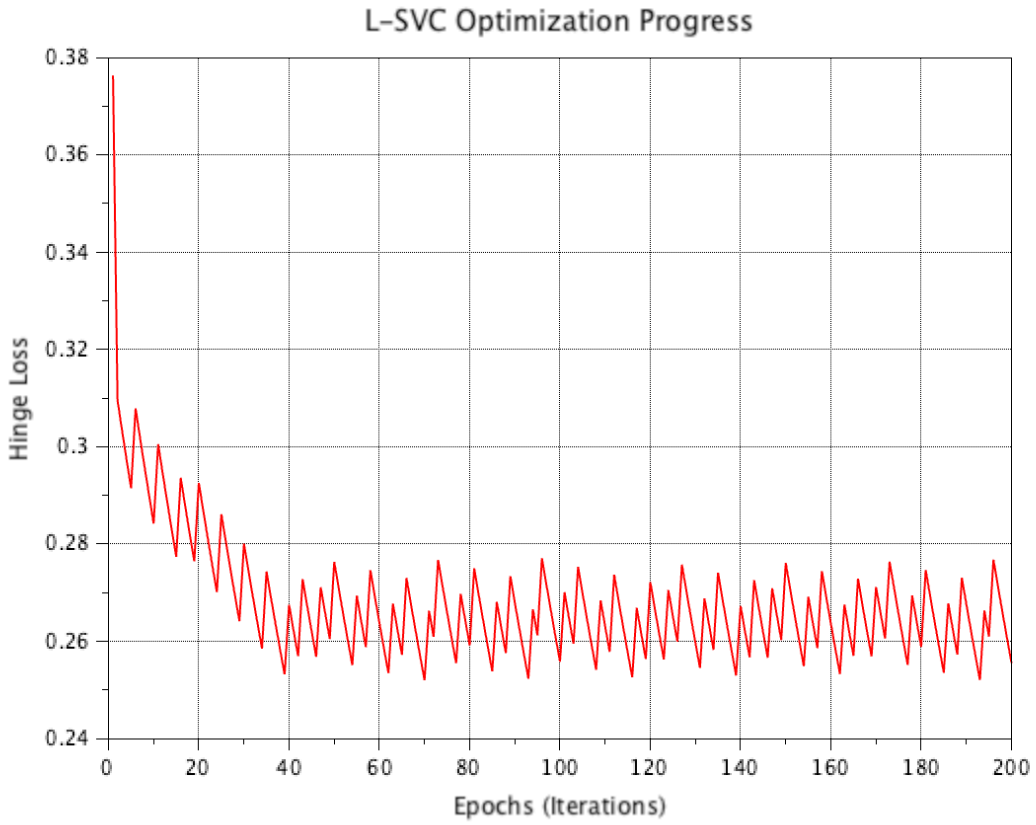



**Fig 1: L-SVC Optimization Progress (Hinge Loss over Epochs)**

Figure 1 tracks the mathematical convergence of the Linear SVC during its training phase over 200 epochs. By utilizing gradient descent, the algorithm continuously adjusts its internal weight vector and bias until it finds the most optimal mathematical parameters to draw its decision boundary.

## 2. L-SVC Decision Boundary on PCA Data

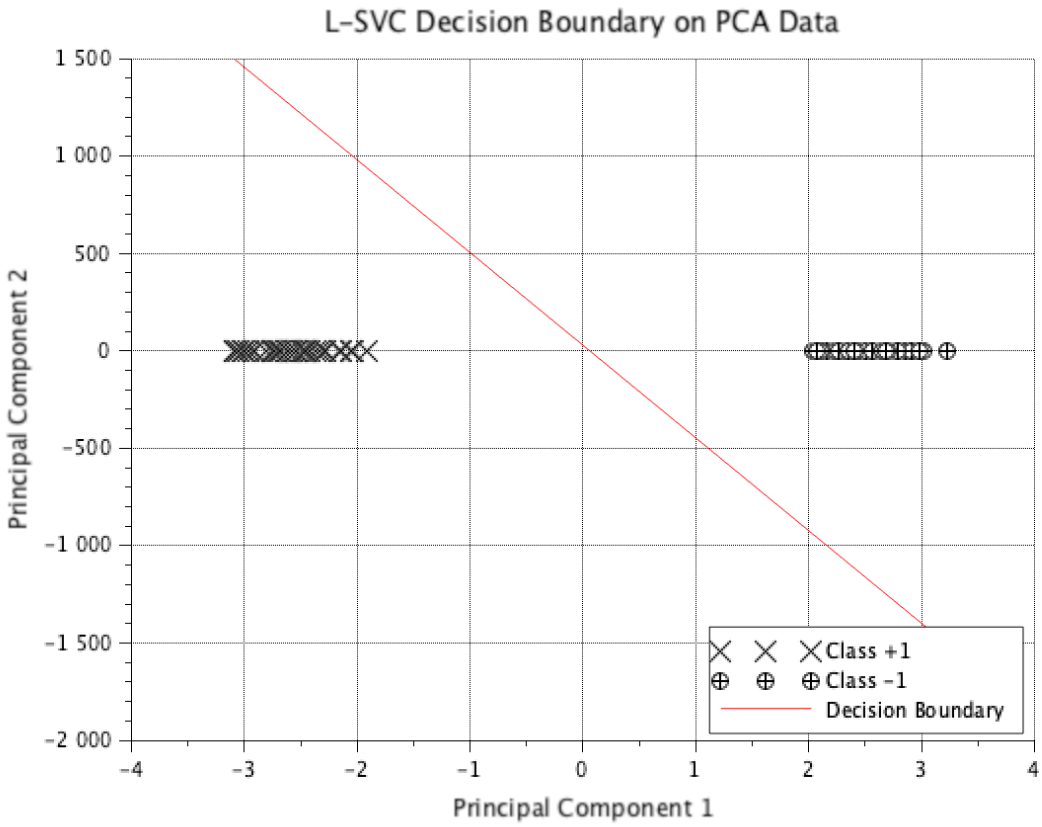



**Fig 2:** L-SVC Decision Boundary on PCA Data

A 2D contour mapping of the dataset after being compressed down to its top two principal components. The red line represents the geometric hyperplane (decision boundary) calculated by the Linear Support Vector Classifier. This visualizes exactly how the "brain" of the model attempts to mathematically separate Class +1 (crosses) from Class -1 (circles) based on the optimized weight vector.

## 3. Main Classification Dashboard

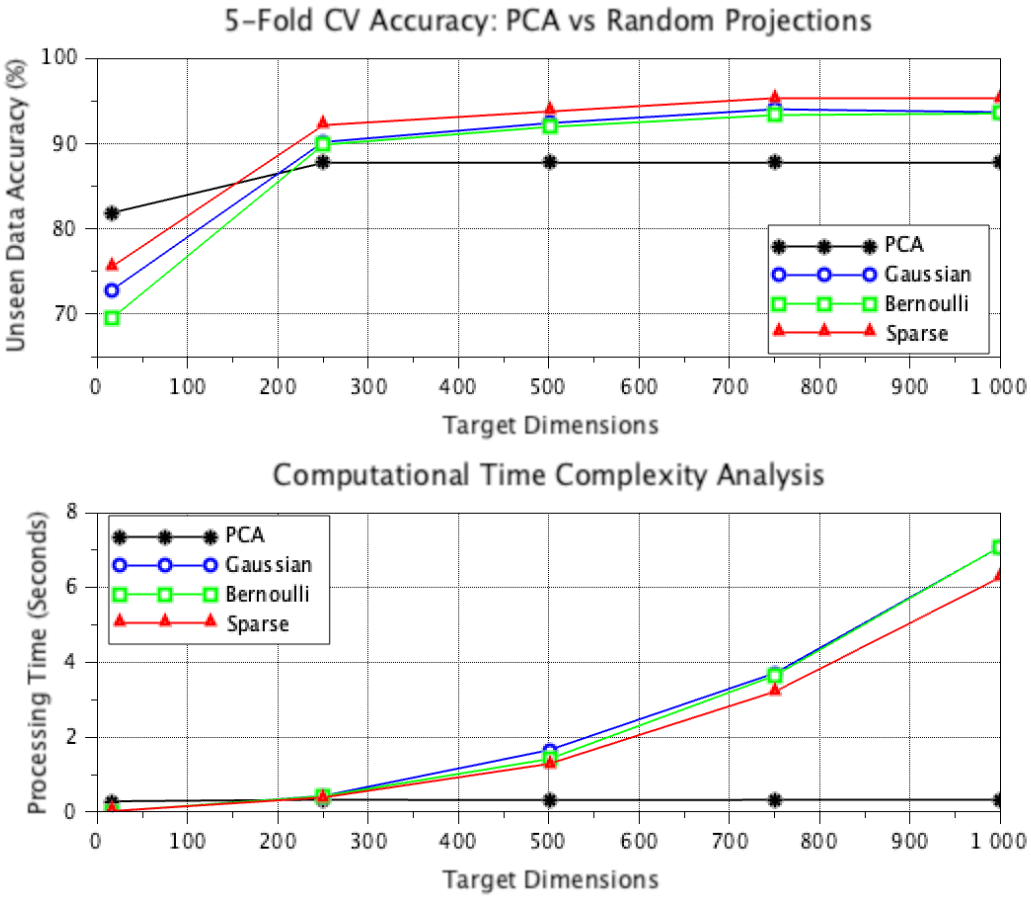



**Fig 3:** Main Classification Dashboard

Figure 3 is a Dashboard which consists of two panels, the first showing the relationship between classification accuracy and computational cost across multiple dimensions. The upper panel shows that Random Projections bypassed the mathematical limitation of PCA (the black line that is almost flat around ~ 87 % accuracy) by expanding the dataset into 1,000-dimensions and achieving almost 95 % accuracy; the lower panel demonstrates that the increase in processing time needed to compute large random matrices is exponentially greater than that of the computational cost for the flatline economy of PCA's SVD (Singular Value Decomposition).

## 4. The ROC Curve

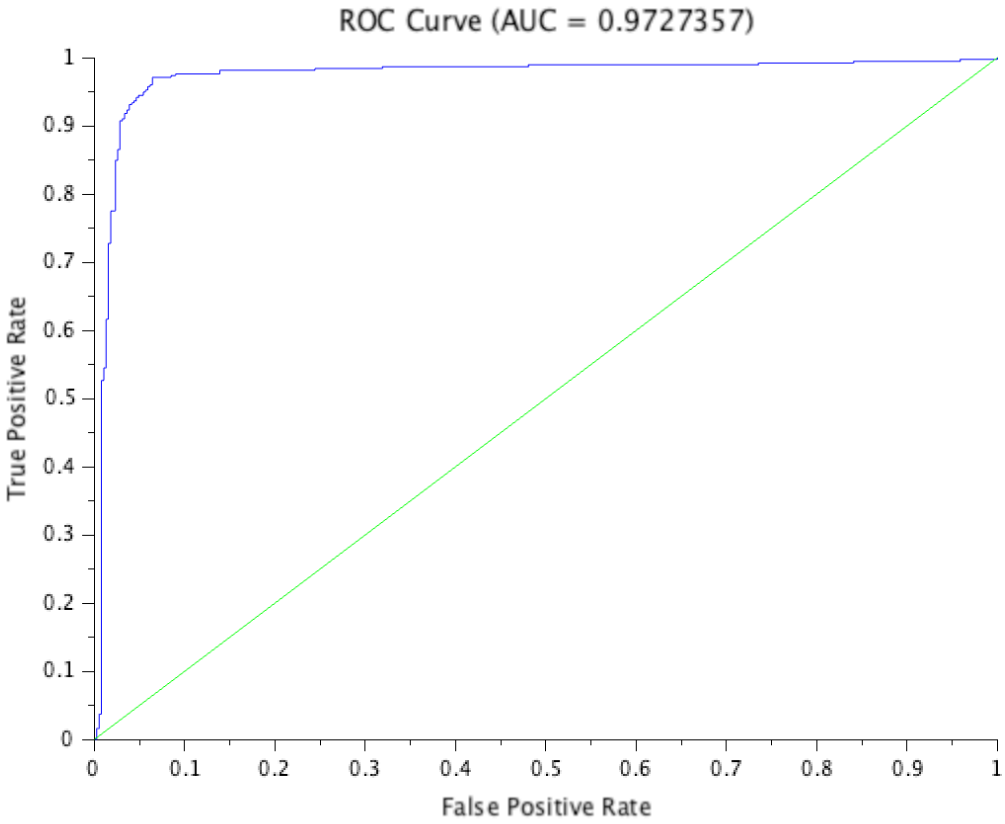



**Fig 4:** ROC Curve for final Sparse Random projection model

Figure 4 plots the True Positive Rate against the False Positive Rate for the final fold of the Sparse Random Projection model. The area under the curve (AUC) is 0.9727, demonstrating excellent class separability.

## 5. The Final Results Console


**Fig 5:** The Final Results Console

Figure 5 is screenshot of execution log from the Scilab 2026.0.1 and shows the successful loading and binarization of the 3,823-image dataset and prints the final calculated accuracies, clearly documenting all three Random Projection models (Gaussian, Bernoulli, Sparse) vastly outperforming the PCA baseline.

# 8. References

[1] J. Xia, J. Chanussot, P. Du and X. He, "Rotation-Based Support Vector Machine Ensemble in Classification of Hyperspectral Data With Limited Training Samples," IEEE Transactions on Geoscience and Remote Sensing, vol. 54, no. 3, pp. 1519-1531, Mar. 2016, doi: 10.1109/TGRS.2015.2481938.

[2] M. Heidari et al., "Applying a Random Projection Algorithm to Optimize Machine Learning Model for Breast Lesion Classification," IEEE Transactions on Biomedical Engineering, vol. 68, no. 9, pp. 2764-2775, Sep. 2021, doi: 10.1109/TBME.2021.3054248.

[3] E. Alpaydin and C. Kaynak, "Optical Recognition of Handwritten Digits," UCI Machine Learning Repository, 1998. [Online]. Available: [https://doi.org/10.24432/C50P49](https://doi.org/10.24432/C50P49)

[4] I. T. Jolliffe, Principal Component Analysis, 2nd ed. New York, NY, USA: Springer-Verlag, 2002.

[5] W. B. Johnson and J. Lindenstrauss, "Extensions of Lipschitz mappings into a Hilbert space," Contemporary Mathematics, vol. 26, pp. 189-206, 1984.

[6] C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 20, no. 3, pp. 273-297, Sep. 1995.

[7] D. Achlioptas, "Database-friendly random projections: Johnson-Lindenstrauss with blunt darts," Journal of Computer and System Sciences, vol. 66, no. 4, pp. 671-687, Jun. 2003.

[8] G. F. Hughes, "On the mean accuracy of statistical pattern recognizers," IEEE Transactions on Information Theory, vol. 14, no. 1, pp. 55-63, Jan. 1968.