Configuration Manual

MSc Research Project
MSc. in Data Analytics

Srushti Wadkar
Student ID: 23201622

School of Computing
National College of Ireland

Supervisor: Prof. Christian Horn

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Srushti Wadkar		
Student ID:	23201622		
Programme:	MSc. in Data Analytics	**Year:**	2024-25
Module:	Research Practicum		
Lecturer:	Christian Horn		
Submission Due Date:	11 August 2025		
Project Title:	Enhancing t-SNE with PCA Initialization and Hybrid Distance Metrics		
Word Count:	1187	**Page Count:**	10

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Srushti Wadkar

Date: 15 September 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Srushti Wadkar
Student ID: 23201622

1 Project Overview

Your first section. Change the header and label to something appropriate.
The paper analyzes the enhancement of the traditional t-Distributed Stochastic Neighbor Embedding (t-SNE) algorithm for the visualization of the high-dimensional data using two data sets, i.e. the Taiwanese Bankruptcy and MNIST (handwritten digits). The standard t-SNE, though good in preserving local neighboring structures, has following limitations:

- Poor reproducibility due to random initialization.
- Ineffective Global structure preservation.

To avoid such limitations the following enhancements were made and planned:

- Improved reproducibility through PCA based initialization.
- Hybrid Distance Metric for Global structure retention.
- Higher dimensional embeddings (5D) for better structural capture or representation.

In order to comprehend how the improvements are compared with the baseline t-SNE, they were tested independently and combined with others. Python 3.12 was used to conduct the studies in a Jupyter Notebook.

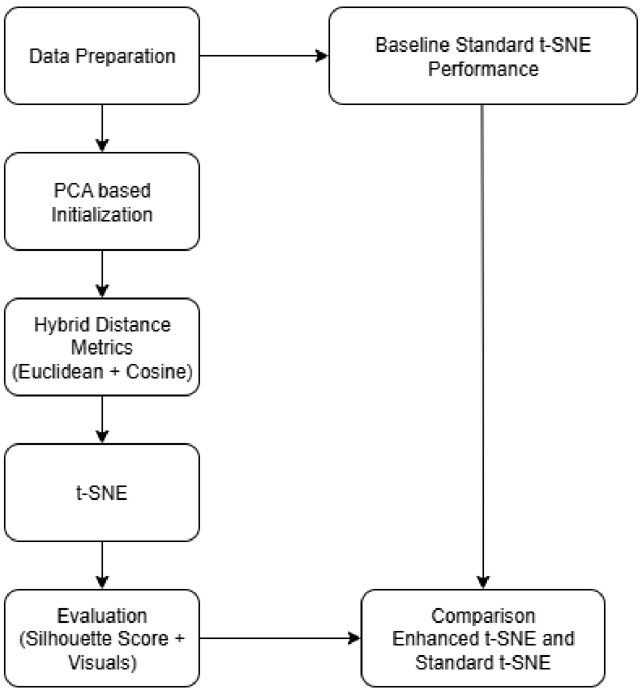

Figure 1: Project Flow.

2 Files and Descriptions

Table 1: Project Files and Descriptions

File Name	Description
mnist_train.csv	MNIST source train data.
sample_df.csv	MNIST 2000 sampled data used in this project.
data.csv	Taiwanese Bankruptcy Source data.
cleaned_data.csv	Taiwanese Bankruptcy VIF analysed cleaned data used in this project.
VIF Analysis.ipnyb	Python code file of VIF Analysis on Bankruptcy data.
Standard_tSNE.ipynb	Python code file of standard t-SNE baseline on both datasets.
Improved_TSNE_Bankruptcy_Data.ipynb	Python code file of enhanced t-SNE on Bankruptcy data.
Improved_TSNE_MNIST_Data.ipynb	Python code file of enhanced t-SNE on MNIST data.
5D_TSNE_V01.ipnyb & 5D_tSNE_V02.ipnyb	Python Code file of 5D embedded enhanced t-SNE on MNIST data.

3 Tools and Environment Setup

- **Programming Language:** Python 3.12
- **Development IDE:** Jupyter Notebook
- **Environment:** Local Machine, CPU execution
- **Libraries Used:**

Table 2: Used Libraries, Versions and Purpose

Library Name	Version	Purpose
Scikit-learn	1.6.1	t-SNE, PCA, Distance Metrics, Evaluation, Feature Normalization
Pandas	2.2.2	Data manipulation, Preparation
Numpy	1.26.4	Data manipulation, Preparation
Matplotlib	3.8.4	Data Visualization
Seaborn	0.13.2	Data Visualization

```
import numpy as np
import pandas as pd
from sklearn.decomposition import PCA
from sklearn.metrics import pairwise_distances
from sklearn.manifold import TSNE
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import silhouette_score
from sklearn.preprocessing import StandardScaler
```

Figure 2: Libraries Used

4 Datasets and Preparations

For this project experiments two different datasets are used, those are as follows:

4.1 MNIST Dataset

- **Source:** The MNIST Dataset is a publicly available and includes picture of handwritten numbers from 0 to 9 in grayscale. (mnist_train.csv)
- **Total Samples:** 60,000 training images.
- **Image Dimensions:** 28×28 pixels.
- **Features:** 784 features are flattening form of 28×28 pictures.
- **Sampling:** 2000 random images of the training set sampled to minimize the cost of computation while t-SNE enhancement. This subset is diverse even with respect to all digit classes. (Sample_df.csv)
- Pandas is used to load and transform it in image format to tabular form (784 features + 1 label).
- **File Format:** csv file. (mnist_train.csv, sample_df.csv)
- **Use in t-SNE enhancement Experiment:**
    - Baseline standard t-SNE evaluation.
    - For 2D and 5D embedded t-SNE enhancement experiments.

```
df = pd.read_csv("mnist_train.csv")

sample_df = df.iloc[:2000]
y = sample_df.iloc[:, 0].values  # labels
X = sample_df.iloc[:, 1:].values  # pixel values

print(X.shape, y.shape)
```

```
(2000, 784) (2000,)
```

```
sample_df.to_csv("sample_df.csv", index = False)
```

Figure 3: MNIST Dataset sampling.

4.2 Taiwanese Bankruptcy Dataset

- **Source**: Taiwanese Bankruptcy Dataset is a publicly available financial records of Taiwanese companies. In order to prevent the multicollinearity among features, a cleaned copy of the dataset is prepared according to Variation Inflation Factor (VIF) analysis.
- **Total Samples**: ~6000 samples with 96 features.
- **Samples after VIF Analysis:** ~6000 samples with 72 features.
- **Target Variables:** Binary Classification (1-Bankrupt and 0-Non-Bankrupt)
- **File Format:** csv file. (data.csv, cleaned_data.csv)
- **Use in t-SNE enhancement Experiment:**
    - Baseline standard t-SNE evaluation.
    - For 2D embedded t-SNE enhancement experiments.

```
#Calculate VIF for each feature in a DataFrame
def calculate_vif(df):
    vif_df = pd.DataFrame()
    vif_df["Feature"] = df.columns
    vif_df["VIF"] = [variance_inflation_factor(df.values, i) for i in range(df.shape[1])]
    return vif_df

# Start with scaled data
current_df = scaled_df.copy()
removed_features = []
threshold = 10

# Iteratively remove features with highest VIF
while True:
    vif_result = calculate_vif(current_df)
    max_vif = vif_result["VIF"].max()
    if max_vif <= threshold:
        break
    else:
        feature_to_remove = vif_result.sort_values("VIF", ascending=False).iloc[0]["Feature"]
        removed_features.append((feature_to_remove, max_vif))
        current_df = current_df.drop(columns=[feature_to_remove])

# summary of removed features
removed_df = pd.DataFrame(removed_features, columns=["Removed Feature", "VIF Value"])
```

Figure 4: VIF Analysis on Bankruptcy Dataset.

5 Baseline t-SNE Implementation

Code file 'Standard_tSNE.ipynb' in both Bankruptcy Data and MNIST Data.

Standard t-SNE was implemented first to observe baseline performance for both datasets. Standard t-SNE Configuration:

```
# standard t-SNE
tsne_standard = TSNE(
    n_components=2,
    metric='euclidean',
    init='random',
    perplexity=30,
)

X_tsne_standard = tsne_standard.fit_transform(X)
score_standard = silhouette_score(X_tsne_standard, y)
```

Figure 5: Baseline t-SNE configuration on both Dataset.

Results of this step were further used as benchmark to evaluate enhancements. Evaluation is done through visuals and Silhouette Score.

```
plt.figure(figsize=(8, 6))
plt.scatter(X_tsne_standard[:, 0], X_tsne_standard[:, 1], c=y, cmap='tab10', s=10)
plt.title(f"Standard t-SNE (Perplexity=30), Silhouette = {score_standard:.4f}")
plt.grid(True)
plt.show()

# silhouette score
print(f"Silhouette Score (Standard t-SNE): {score_standard:.4f}")
```

Figure 6: Baseline t-SNE Evaluation.

6 Enhanced t-SNE Configuration

Code file 'Improved_TSNE_Bankruptcy_Data.ipnyb' for Bankruptcy Data and 'Improved_TSNE_MNIST_Data.ipnyb' for MNIST Data.

Code files '5D_TSNE_V01.ipnyb' and '5D_TSNE_V02.ipnyb' for t-SNE 5D embedding for MNIST Data.

6.1 PCA Initialization (2D)

For deterministic start point in t-SNE, random initialization is replaced by PCA initialization. Default PCA and t-SNE embedding is 2D (n_components=2), similar is used for first experiment, applied to both Datasets (Figure). Further, it is applied to t-SNE through 'init' parameter as follows (Figure).

```
pca = PCA(n_components=2)
X_pca_init = pca.fit_transform(X_scaled)

X_pca_init.shape
```

```
tsne = TSNE(
    n_components=2,
    metric='precomputed',
    init=X_pca_init,
    perplexity=30,
    random_state=42
)

X_tsne = tsne.fit_transform(hybrid_dist)
```

Figure 7: PCA Initialization Configuration on both Datasets.

Evaluation is done through Visualizations across multiple runs.

6.2 Hybrid Distance Metric

Hybrid Distance Metric is combined Euclidean and Cosine distances.
Formula Used:
$Hybrid\ Distance = 0.5 \times Euclidean_{norm} + 0.5 \times (1 - Cosine_{norm})$

```
# Euclidean distance matrix
euclidean_dist = pairwise_distances(X, metric='euclidean')

# Cosine similarity to distance (1 - similarity)
cosine_sim = 1 - pairwise_distances(X, metric='cosine')

# Normalize both matrices to [0, 1]
euclidean_norm = (euclidean_dist - euclidean_dist.min()) / (euclidean_dist.max() - euclidean_dist.min())
cosine_norm = (cosine_sim - cosine_sim.min()) / (cosine_sim.max() - cosine_sim.min())

hybrid_dist = 0.5 * euclidean_norm + 0.5 * (1 - cosine_norm)
```

Figure 8: Hybrid Distance Metrics Calculations.

This Hybrid Distance Metric is passed to t-SNE with 'metric='precomputed''. Applied in similar way to both Datasets for all embeddings (2D or 5D).

```python
tsne = TSNE(
    n_components=2,
    metric='precomputed',
    init=X_pca_init,
    perplexity=30,
    random_state=42
)

X_tsne = tsne.fit_transform(hybrid_dist)
```

Figure 9: Hybrid Distance in t-SNE.

Evaluation is done through Visualizations and Silhouette Scores.

6.3 5D Embeddings (MNIST Data Only)

For MNIST Data, 5D embedding is experimented (n_components=5). Note: PCA and t-SNE embeddings should be similar to apply PCA as initialization to t-SNE.
'method='exact'' is additional parameter because of algorithmic shortcomings of Barnes-Hut with higher dimensions (Required for dimensions more than 3D).

```python
# PCA with 5D
pca_5d = PCA(n_components=5)
X_pca_5d = pca_5d.fit_transform(X)
```

```python
# Run t-SNE with 5D output
tsne_5d = TSNE(
    n_components=5,
    init=X_pca_5d,
    metric='precomputed',
    method='exact',
    perplexity=30,
    random_state=42
)

X_tsne_5d = tsne_5d.fit_transform(hybrid_dist)
```

Figure 10: 5D Embedding in enhanced t-SNE for MNIST Data.

Evaluated using Silhouette Scores.

6.4 Perplexity Experiments

Explored set of perplexity values (5, 10, 20, 30, 40, 50) on whole enhanced t-SNE algorithm to observe effects on Silhouette Scores and cluster compactness.
Key Findings:
2D embeddings showed variations across perplexities.
5D embeddings remained stable across all perplexity values.

```
def run_tsne_perplexities(perplexities, hybrid_dist, init, labels):
    results = []
    for perp in perplexities:
        tsne = TSNE(
            n_components=2,
            metric='precomputed',
            init=init,
            perplexity=perp
        )
        X_tsne = tsne.fit_transform(hybrid_dist)
        score = silhouette_score(X_tsne, labels)
        results.append((perp, score))

        plt.figure(figsize=(10, 8))
        sns.scatterplot(x=X_tsne[:, 0], y=X_tsne[:, 1], hue=y, palette='tab10', s=60, alpha=0.6, edgecolor='k', linewidth=0.5)
        for label in np.unique(y):
            idx = y == label
            x_text = np.median(X_tsne[idx, 0])
            y_text = np.median(X_tsne[idx, 1])
            plt.text(x_text, y_text, str(label), fontsize=30, weight='bold',
                     horizontalalignment='center', verticalalignment='center', color='black',
                     bbox=dict(facecolor='white', edgecolor='black', boxstyle='round,pad=0.1'))

        plt.title(f"Perplexity = {perp}, Silhouette = {score:.3f}")
        plt.xlabel("Component 1")
        plt.ylabel("Component 2")
        plt.legend(title='Digit', bbox_to_anchor=(1.05, 1), loc='upper left')
        plt.grid(True)
        plt.tight_layout()
        plt.show()

    return results


perplexities = [5, 10, 20, 30, 40, 50]
results = run_tsne_perplexities(perplexities, hybrid_dist, X_pca_init, y)


for perp, score in results:
    print(f"Perplexity: {perp}, Silhouette Score: {score:.4f}")
```

Figure 11: Enhanced t-SNE experimented at various perplexity.

Applied in similar way to both datasets across all embeddings (2D or 5D).

7 Evaluation Methods

Table 3: Evaluation Methods

Criterion	Method	Notes
Cluster Quality and Global Structure	Silhouette Score and Visual Inspection	Higher Score equal better separated clusters
Reproducibility	Multiple runs check on visuals	PCA initialization showed consistent output, without PCA initialization variation in outputs.

8 Sample Results Summary

Following are the visuals of experiments carried out:

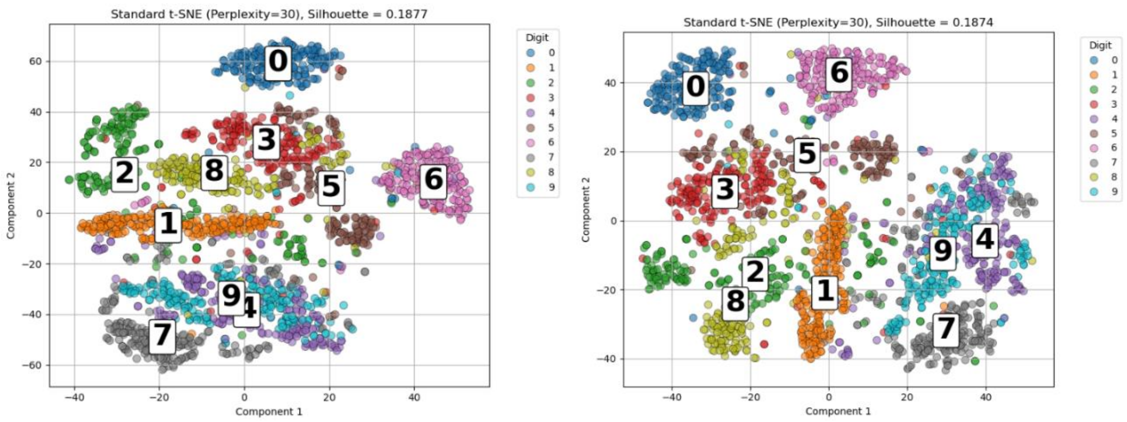

Figure 12: Baseline t-SNE Visuals of MNIST Dataset.

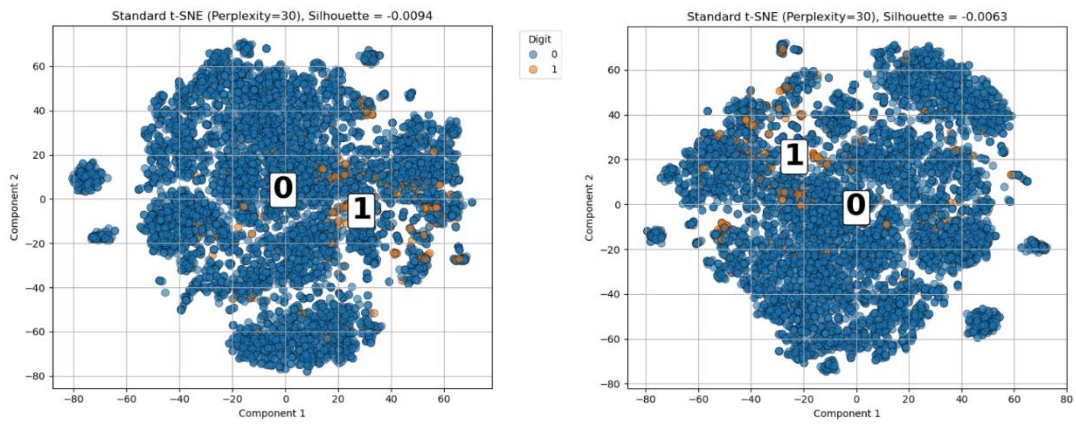

Figure 13: Baseline t-SNE Visuals of Bankruptcy Dataset.

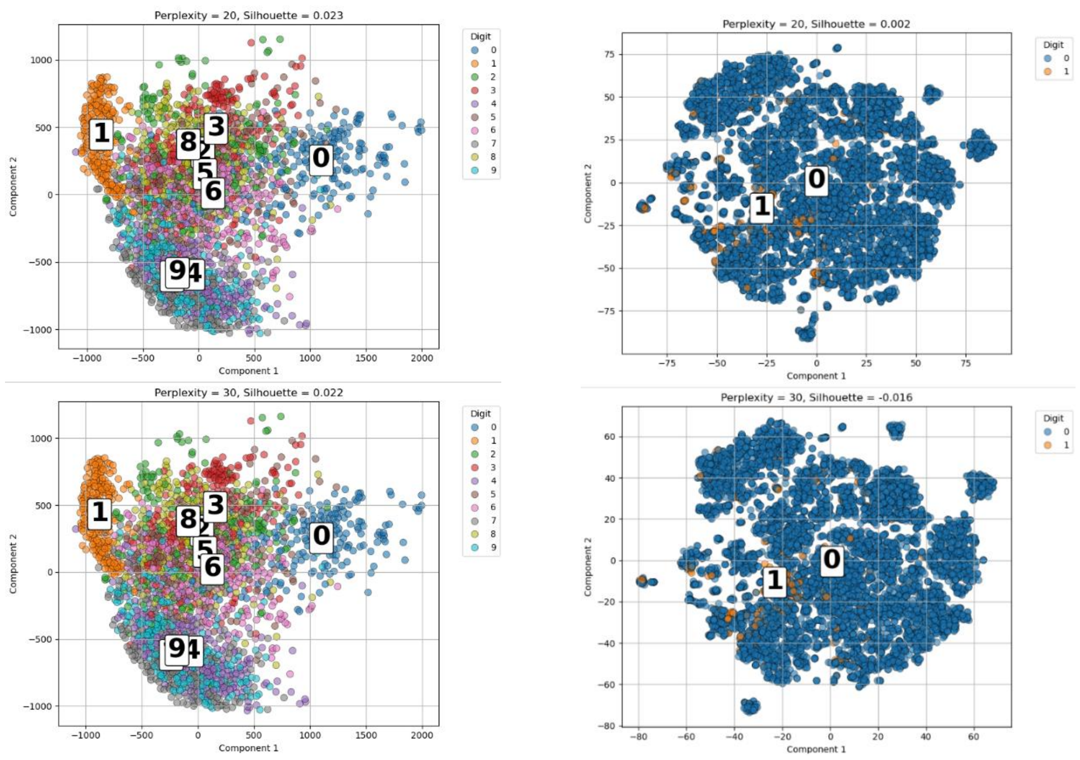

Figure 14: MNIST (Left) and Bankruptcy (Right) Data Visuals with PCA Initialization

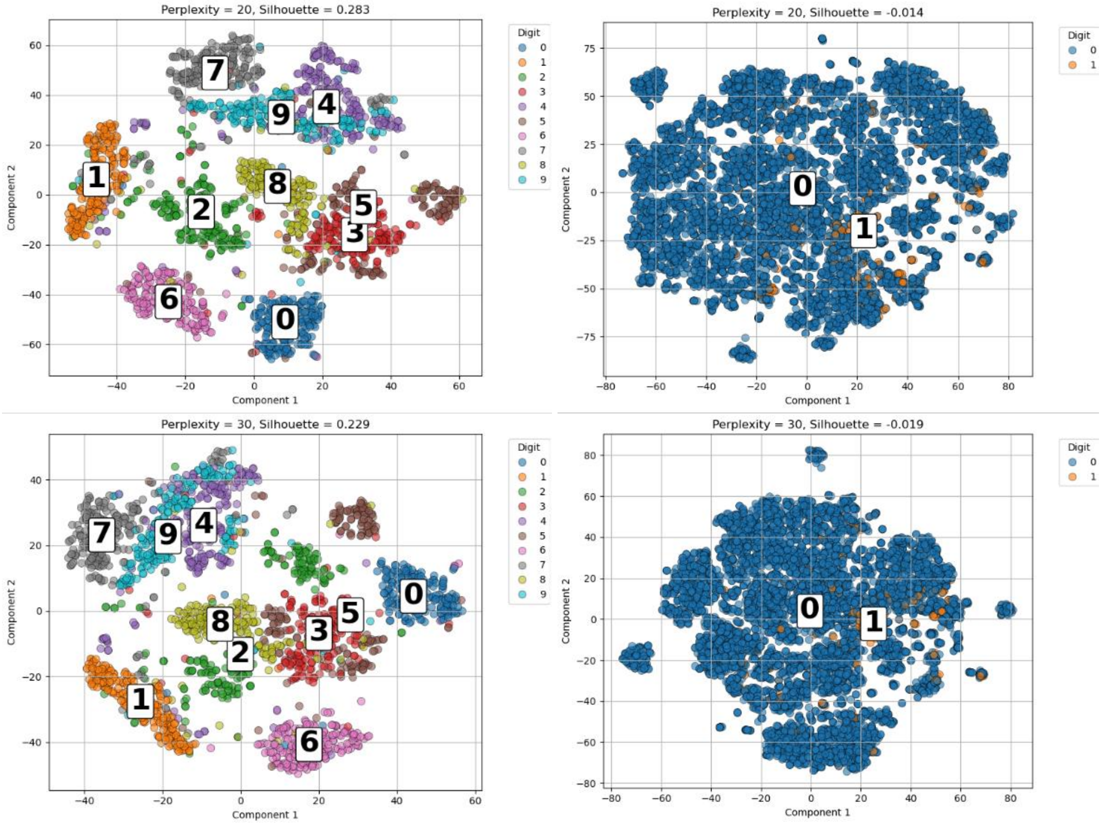

Figure 15: MNIST (Left) and Bankruptcy (Right) Data Visuals without PCA Initialization

Following are the Silhouette Score of experiments carried out:

Table 4: Silhouette Score Results by Experiments

Dataset	**Method**	**Silhouette Score (Best of all Perplexity)**
MNIST Data (2D)	Baseline Performance	0.1878
MNIST Data (2D)	Hybrid Distance	0.2846
MNIST Data (2D)	PCA + Hybrid Distance	0.0231
MNIST Data (5D)	PCA + Hybrid Distance	0.0734
Bankruptcy Data (2D)	Baseline Performance	-0.0072
Bankruptcy Data (2D)	Hybrid Distance	-0.0120
Bankruptcy Data (2D)	PCA + Hybrid Distance	0.0025

References

2.2. Manifold learning — scikit-learn 0.22.2 documentation. Available at: https://scikit-learn.org/stable/modules/manifold.html.

3D scatterplot — Matplotlib 3.7.1 documentation. Available at: https://matplotlib.org/stable/gallery/mplot3d/scatter3d.html.

Awan, A.A. 2023. Python t-SNE with Matplotlib. Available at: https://www.datacamp.com/tutorial/introduction-t-sne.

Contributor, Stats.S. 2024. Dimensionality reduction and precomputed distance matrix. Available at: https://stats.stackexchange.com/questions/644469/dimensionality-reduction-and-precomputed-distance-matrix.

Dato-on, D. 2018. MNIST in CSV. Available at: https://www.kaggle.com/datasets/oddrationale/mnist-in-csv?resource=download.

HugoLasticot. 2016. Cosine similarity TSNE in sklearn.manifold. Available at: https://stackoverflow.com/questions/36545434/cosine-similarity-tsne-in-sklearn-manifold.

JADHAV, P. 2025. How Machines Measure Similarity? From Cosine to Mahalanobis! Available at: https://pandurang-jadhav.medium.com/how-machines-measure-similarity-from-cosine-to-mahalanobis-bdfeebf37a99.

misha-pyshark. 2020. Initialization of tSNE with PCA, allow for “auto” instead of “random.” Available at: https://github.com/scikit-learn/scikit-learn/issues/16986.

Preserving global structure — openTSNE 1.0.0 documentation. 2015. Available at: https://opentsne.readthedocs.io/en/latest/examples/03_preserving_global_structure/03_preserving_global_structure.html.

scikit-learn developers. 2014. sklearn.manifold.TSNE — scikit-learn 0.21.3 documentation. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html.

seaborn.scatterplot — seaborn 0.11.1 documentation. Available at: https://seaborn.pydata.org/generated/seaborn.scatterplot.html.

UCI Machine Learning Repository. Available at: https://archive.ics.uci.edu/dataset/572/taiwanese+bankruptcy+prediction.