

```

# 1. Import required libraries
import numpy as np
import matplotlib.pyplot as plt

from sklearn.datasets import load_iris
from sklearn.preprocessing import StandardScaler
from sklearn.manifold import TSNE, trustworthiness
from sklearn.metrics import silhouette_score

# 2. Load dataset (Iris as an example)
# X = features, y = class labels
X, y = load_iris(return_X_y=True)

# 3. Standardize the data
# IMPORTANT: t-SNE is very sensitive to feature scales
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# 4. Initialize and run t-SNE
# n_components = 2 for 2D visualization
# perplexity controls local neighborhood size
# metric defines distance calculation
tsne = TSNE(
    n_components=2,
    perplexity=30,
    learning_rate=200,
    n_iter=1000,
    metric="mahalanobis", # try "cosine" for comparison
    random_state=42
)

# Fit t-SNE and transform data
X_tsne = tsne.fit_transform(X_scaled)

# 5. Compute evaluation metrics

# 5.1 Trustworthiness (best metric for t-SNE)
# Measures how well local neighborhoods are preserved
trust = trustworthiness(X_scaled, X_tsne, n_neighbors=5)
print(f"Trustworthiness score: {trust:.4f}")

# 5.2 Silhouette score (optional, label-dependent)
# Measures how well classes are separated in embedding
sil_score = silhouette_score(X_tsne, y)
print(f"Silhouette score: {sil_score:.4f}")

# 6. Visualization of t-SNE results
plt.figure(figsize=(8, 6))

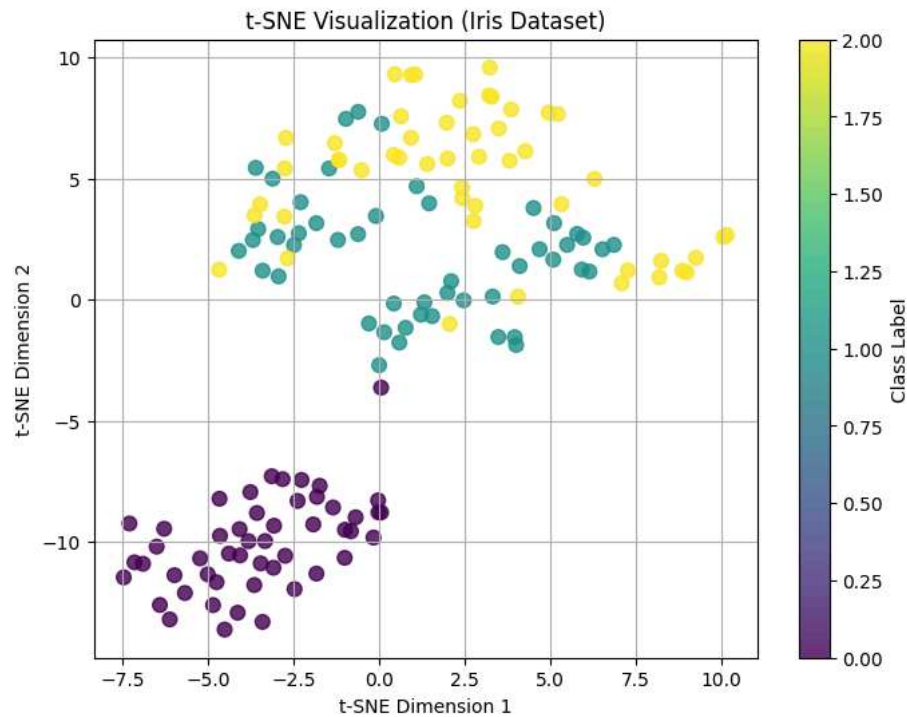
scatter = plt.scatter(
    X_tsne[:, 0], # t-SNE dimension 1
    X_tsne[:, 1], # t-SNE dimension 2
    c=y, # color by class label
    cmap="viridis",
    s=60,
    alpha=0.8
)

# Add color bar and labels
plt.colorbar(scatter, label="Class Label")
plt.title("t-SNE Visualization (Iris Dataset)")
plt.xlabel("t-SNE Dimension 1")
plt.ylabel("t-SNE Dimension 2")
plt.grid(True)

# Display plot
plt.show()

```

```
/usr/local/lib/python3.12/dist-packages/sklearn/manifold/_t_sne.py:1164: FutureWarning: 'n_iter' was renamed to 'max_iter' in ve
warnings.warn(
Trustworthiness score: 0.9361
Silhouette score: 0.3351
```



```
import pandas as pd
url = "https://raw.githubusercontent.com/pandas-dev/pandas/main/pandas/tests/io/data/csv/tips.csv" # Corrected URL to an existi
data = pd.read_csv(url)
print(data.head())
data.info()
data.describe()
```

```
total_bill  tip    sex smoker  day    time  size
0      16.99  1.01  Female    No  Sun  Dinner     2
1      10.34  1.66   Male    No  Sun  Dinner     3
2      21.01  3.50   Male    No  Sun  Dinner     3
3      23.68  3.31   Male    No  Sun  Dinner     2
4      24.59  3.61  Female    No  Sun  Dinner     4
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 244 entries, 0 to 243
```

```
Data columns (total 7 columns):
```

```
#    Column    Non-Null Count  Dtype
---  -
0    total_bill  244 non-null       float64
1    tip         244 non-null       float64
2    sex         244 non-null       object
3    smoker      244 non-null       object
4    day         244 non-null       object
5    time        244 non-null       object
6    size        244 non-null       int64
```

```
dtypes: float64(2), int64(1), object(4)
```

```
memory usage: 13.5+ KB
```

	total_bill	tip	size
count	244.000000	244.000000	244.000000
mean	19.785943	2.998279	2.569672
std	8.902412	1.383638	0.951100
min	3.070000	1.000000	1.000000
25%	13.347500	2.000000	2.000000
50%	17.795000	2.900000	2.000000
75%	24.127500	3.562500	3.000000
max	50.810000	10.000000	6.000000

```
import pandas as pd
```

```
url = "https://raw.githubusercontent.com/pandas-dev/pandas/main/pandas/tests/io/data/csv/iris.csv"
data = pd.read_csv(url)
print(data.head())
data.info()
data.describe()
```

```
   SepalLength  SepalWidth  PetalLength  PetalWidth      Name
0         5.1         3.5         1.4         0.2  Iris-setosa
1         4.9         3.0         1.4         0.2  Iris-setosa
2         4.7         3.2         1.3         0.2  Iris-setosa
3         4.6         3.1         1.5         0.2  Iris-setosa
4         5.0         3.6         1.4         0.2  Iris-setosa
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 150 entries, 0 to 149
```

```
Data columns (total 5 columns):
```

#	Column	Non-Null Count	Dtype
0	SepalLength	150 non-null	float64
1	SepalWidth	150 non-null	float64
2	PetalLength	150 non-null	float64
3	PetalWidth	150 non-null	float64
4	Name	150 non-null	object

```
dtypes: float64(4), object(1)
```

```
memory usage: 6.0+ KB
```

	SepalLength	SepalWidth	PetalLength	PetalWidth
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000