EXPERIMENT NO 10

AIM: To perform t-SNE (t-Distributed Stochastic Neighbor Embedding) on the given data.

SOFTWARE USED: Jupyter Notebook

THEORY:

t-Distributed Stochastic Neighbor Embedding (t-SNE):

t-SNE is a nonlinear technique used for visualizing high-dimensional data in a lower-dimensional space, typically 2D or 3D. It focuses on preserving the local structure of the data by modeling pairwise similarities between data points in both the high-dimensional and low-dimensional spaces. It does this by representing each high-dimensional data point as a probability distribution in the low-dimensional space and minimizing the Kullback-Leibler divergence between the distributions in the high-dimensional and low-dimensional spaces.

Pros of t-SNE:

Non-Linearity: t-SNE can capture nonlinear relationships between variables, making it suitable for complex datasets.

Preservation of Local Structure: t-SNE excels at preserving the local structure of the data, making it particularly useful for visualizing clusters or groups of similar data points.

Robustness to Noise: t-SNE is robust to noise and can effectively handle datasets with noisy or ambiguous patterns.

Cons of t-SNE:

Computational complexitivity :t-SNE can be computationally expensive, especially for large datasets, due to its iterative optimization process.

Loss of Global Structure: While t-SNE preserves local structure well, it may distort global relationships between data points, leading to misleading interpretations of the overall dataset.

Comparison:

Linearity vs. Nonlinearity: PCA assumes linear relationships between variables, while t-SNE can capture nonlinear relationships.

Global vs. Local Structure: PCA preserves global structure better, while t-SNE focuses on preserving local structure.

Interpretability vs. Visualization: PCA provides interpretable principal components, while t-SNE is primarily used for visualization and exploration of high-dimensional data.

Working of t-sne:

Initialization: The t-SNE algorithm begins by initializing the low-dimensional embeddings randomly. These embeddings represent the data points in the lower-dimensional space where the visualization will be constructed.

Compute Pairwise Similarities: Next, t-SNE calculates the pairwise similarities between data points in the high-dimensional space. Typically, Gaussian kernels are used to measure the similarity between points. This step establishes the foundation for understanding the relationships between data points in their original highdimensional space.

Compute Conditional Probabilities: After computing the pairwise similarities, t-SNE transforms these similarities into conditional probabilities in the low-dimensional space. These probabilities indicate how likely it is for two points to select each other as neighbors in the reduced-dimensional representation. A Student's tdistribution is often utilized for this purpose, allowing for the preservation of both nearby and distant relationships.

Optimization Objective: The main goal of t-SNE is to minimize the difference between the conditional probabilities in the high-dimensional space and those in the low-dimensional space. This is achieved by minimizing the Kullback-Leibler (KL) divergence between the two sets of probabilities. Minimizing the KL divergence ensures that the relationships between data points are accurately preserved during dimensionality reduction.

Gradient Descent: To minimize the KL divergence, t-SNE employs an optimization technique such as gradient descent. It iteratively adjusts the low-dimensional embeddings to better match the conditional probabilities of the high-dimensional space. Through these adjustments, t-SNE effectively refines the embeddings to capture the underlying structure of the data in the lower-dimensional visualization.

Iterations and Convergence: The optimization process in t-SNE is performed iteratively. It continues until convergence is achieved or until a predefined number of iterations is reached. During each iteration, the algorithm refines the embeddings to progressively improve the alignment between the high-dimensional and low-dimensional probabilities. Convergence indicates that the algorithm has found a stable representation of the data in the reduced-dimensional space.

Final Embeddings: Once the optimization process is complete, t-SNE produces the final low-dimensional embeddings. These embeddings represent the data in a lower-dimensional space, where the relationships between data points are preserved as faithfully as possible. The resulting visualization can then be utilized for

exploratory data analysis, clustering, or any other tasks where understanding the underlying structure of the data is essential.

CONCLUSION:

In conclusion, t-SNE is a powerful technique for visualizing high-dimensional data by preserving local structures. Through an iterative optimization process, it effectively reduces the dimensionality while maintaining the relationships between data points. However, its computational complexity and potential distortion of global structures should be considered when interpreting the results. Overall, t-SNE provides valuable insights into the underlying structure of complex datasets, making it a valuable tool for exploratory data analysis and visualization.

In [2]:

```
!pip install bioinfokit
```

```
Collecting bioinfokit
  Downloading bioinfokit-2.1.3.tar.gz (87 kB)
Requirement already satisfied: pandas in c:\users\hp\anaconda3\lib\sit
e-packages (from bioinfokit) (1.3.4)
Requirement already satisfied: numpy in c:\users\hp\anaconda3\lib\site
-packages (from bioinfokit) (1.20.3)
Requirement already satisfied: matplotlib in c:\users\hp\anaconda3\lib
\site-packages (from bioinfokit) (3.4.3)
Requirement already satisfied: scipy in c:\users\hp\anaconda3\lib\site
-packages (from bioinfokit) (1.7.1)
Requirement already satisfied: scikit-learn in c:\users\hp\anaconda3\l
ib\site-packages (from bioinfokit) (1.4.2)
Requirement already satisfied: seaborn in c:\users\hp\anaconda3\lib\si
te-packages (from bioinfokit) (0.11.2)
Collecting matplotlib-venn
  Downloading matplotlib_venn-0.11.10-py3-none-any.whl (33 kB)
Collecting tabulate
  Downloading tabulate-0.9.0-py3-none-any.whl (35 kB)
Requirement already satisfied: statsmodels in c:\users\hp\anaconda3\li
b\site-packages (from bioinfokit) (0.12.2)
Collecting textwrap3
  Downloading textwrap3-0.9.2-py2.py3-none-any.whl (12 kB)
Collecting adjustText
  Downloading adjustText-1.1.1-py3-none-any.whl (11 kB)
Requirement already satisfied: cycler>=0.10 in c:\users\hp\anaconda3\l
ib\site-packages (from matplotlib->bioinfokit) (0.10.0)
Requirement already satisfied: pillow>=6.2.0 in c:\users\hp\anaconda3
\lib\site-packages (from matplotlib->bioinfokit) (8.4.0)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\hp\anacond
a3\lib\site-packages (from matplotlib->bioinfokit) (3.0.4)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\hp\ana
conda3\lib\site-packages (from matplotlib->bioinfokit) (2.8.2)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\hp\anacon
da3\lib\site-packages (from matplotlib->bioinfokit) (1.3.1)
Requirement already satisfied: six in c:\users\hp\anaconda3\lib\site-p
ackages (from cycler>=0.10->matplotlib->bioinfokit) (1.16.0)
Requirement already satisfied: pytz>=2017.3 in c:\users\hp\anaconda3\l
ib\site-packages (from pandas->bioinfokit) (2021.3)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\hp\ana
conda3\lib\site-packages (from scikit-learn->bioinfokit) (2.2.0)
Requirement already satisfied: joblib>=1.2.0 in c:\users\hp\anaconda3
\lib\site-packages (from scikit-learn->bioinfokit) (1.4.0)
Requirement already satisfied: patsy>=0.5 in c:\users\hp\anaconda3\lib
\site-packages (from statsmodels->bioinfokit) (0.5.2)
Building wheels for collected packages: bioinfokit
  Building wheel for bioinfokit (setup.py): started
  Building wheel for bioinfokit (setup.py): finished with status 'don
e'
  Created wheel for bioinfokit: filename=bioinfokit-2.1.3-py3-none-an
y.whl size=59091 sha256=e18e7e1779b0616850e93417ceebf87d6b5b40d36f5bdc
e6bb52c10838de0d65
  Stored in directory: c:\users\hp\appdata\local\pip\cache\wheels\10\0
```

```
4\be\a31133d287facde61ce7ce52667b5e1f6bfa2ebe5d4f5e86f8
Successfully built bioinfokit
Installing collected packages: textwrap3, tabulate, matplotlib-venn, a
djustText, bioinfokit
Successfully installed adjustText-1.1.1 bioinfokit-2.1.3 matplotlib-ve
```

```
from pandas import read_csv
import pandas as pd
from sklearn.manifold import TSNE

from bioinfokit.visuz import cluster
```

```
filename='TSNE_data.csv'
dataframe=pd.read_csv(filename)
dataframe.head()
```

```
array=dataframe.values
X=array[:,1:]
Y=array[:,0]
```

```
from bioinfokit.visuz import cluster
data_tsne=TSNE(n_components=2).fit_transform(X)
cluster.tsneplot(score=data_tsne)
```

```
nn-0.11.10 tabulate-0.9.0 textwrap3-0.9.2
```

In [*]:

In [*]:

In [6]:

In [1]:

In [9]:

In [ ]:

In [10]:

```
---------------------------------------------------------------------
----
NameError                                 Traceback (most recent call
last)
~\AppData\Local\Temp/ipykernel_11764/1835308856.py in <module>
 1 from bioinfokit.visuz import cluster
```

```
----> 2 data_tsne=TSNE(n_components=2).fit_transform(X)
3 cluster.tsneplot(score=data_tsne) NameError: name
'TSNE' is not defined
```

```
from bioinfokit.visuz import cluster
data_tsne=TSNE(n_components=2).fit_transform(x)
cluster.tsneplot(score=data_tsne)
```

```
---------------------------------------------------------------------
----
NameError                                 Traceback (most recent call
last)
~\AppData\Local\Temp/ipykernel_11632/1422183574.py in <module>
 1 from bioinfokit.visuz import cluster
----> 2 data_tsne=TSNE(n_components=2).fit_transform(x)
 3 cluster.tsneplot(score=data_tsne)

NameError: name 'x' is not defined
```

In [11]:

In [12]:

```
from bioinfokit.visuz import cluster
data_tsne=TSNE(n_components=2).fit_transform(X)
cluster.tsneplot(score=data_tsne)
```

Out[12]:

```
27962  ,  -9.46284  ],
...,
```

```
color_class=dataframe["diagnosis"].to_numpy()
cluster.tsneplot(score=data_tsne, colorlist=color_class, legendpos='uppe
```

```
data_tsne
```

```
array([[ 42.711285 , -12.072052 ],
       [ 42.87602  ,  -9.3755665],
       [ 22.634014 ,  -2.7077618],
       [ 40.747253 ,  -9.473898 ],
       [-37.975258 , -25.647676 ]], dtype=float32)
```

In [ ]:

```
[
3
8
.
```