Principle Component Analysis

(a) What it does:

Principal Component Analysis, or PCA, is a dimensionality reduction method often used to reduce the dimensionality of large data sets by transforming a large set of variables into a smaller one that still contains most of the information from the large set.

Reducing the number of variables in a data set naturally comes at the expense of precision, but the trick to dimensionality reduction is to trade a bit of precision for simplicity. Because smaller data sets are easier to examine and visualize, and analyzing the data is much easier and faster for machine learning algorithms with no extra variables to process.

To summarize, the idea behind PCA is simple – to reduce the number of variables in a data set while preserving as much information as possible.

(b) How it does:

1. Standardize the range of continuous initial variables.
2. Calculate the covariance matrix to identify correlations.
3. Calculate the eigenvectors and eigenvalues of the covariance matrix to identify the principal components.
4. Create a feature vector to decide which principal components to keep.
5. Data retyping along principal component axes.

Step1: Standardization:

The goal of this step is to standardize the range of continuous initial variables so that each contributes equally to the analysis.

More specifically, the reason it is critical to perform standardization before PCA is that PCA is quite sensitive to the variances of the initial variables. This means that if there are large differences between the ranges of the initial variables, those variables with larger ranges will dominate those with small ranges (For example, a variable that ranges between 0 and 100 will dominate a variable that ranges between 0 and 1) , which will lead to biased results. Thus, transforming the data into comparable scales can avoid this problem.

Step 2: Covariance Matrix Computation:

The goal of this step is to understand how the variables of the input data set differ from each other from the mean, or in other words to find out if there is any relationship between them. Because sometimes the variables are highly correlated in such a way that they contain redundant information. So, to identify these correlations, we calculate the covariance matrix.

Step 3: Compute the Eigenvector and Eigenvalues of the Covariance Matrix to identify the Principal Components:

Eigenvectors and Eigen numbers are linear algebra concepts that we need to calculate from the covariance matrix to determine the principal components of the data.

Step 4: Feature Vector:

As we saw in the previous step, computing the eigenvectors and sorting them by their eigenvalues in descending order allows us to find the principal components in order of significance. In this step, we decide whether to keep all these components or discard the less significant ones (with low eigenvalues) and create a vector matrix with the rest, which we call the Feature vector.

Step 5: Recast the data along the principal component axis:

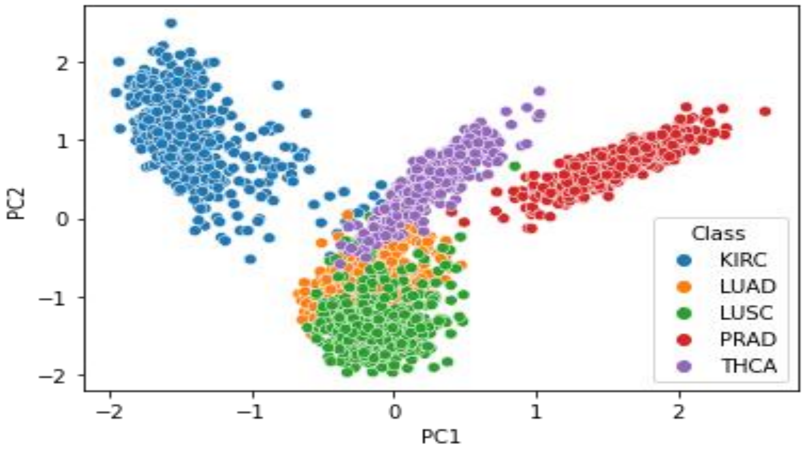
In the previous steps, apart from standardization, you do not make any changes to the data, you only select the principal components and create a feature vector, but the input data set always remains in the original axes (i.e., the initial variables).

(c) Application:

Image compression, Facial recognition, Neuroscience are some of the applications of PCA.

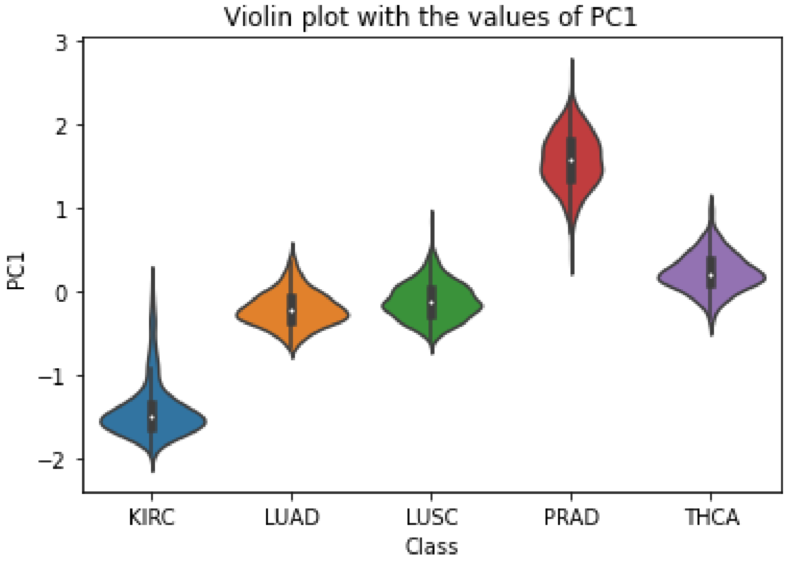
RESULTS for PCA

(A) PCA Visualization



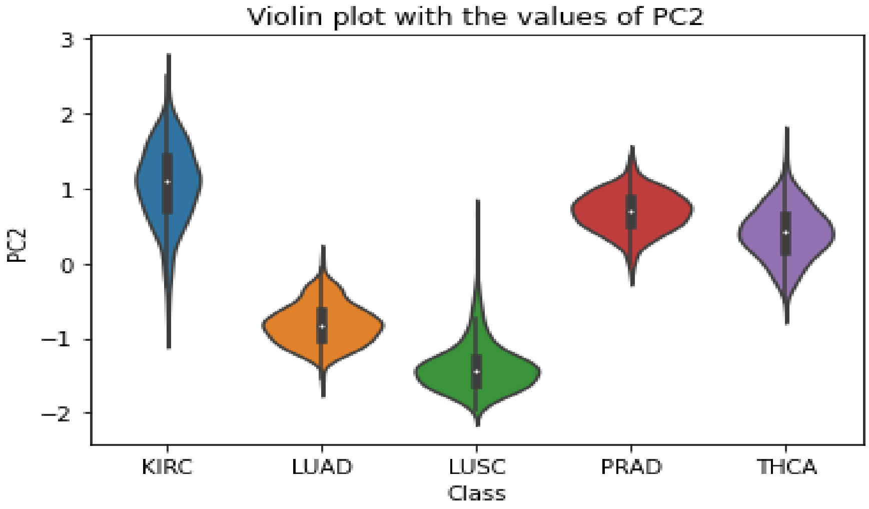
(B) On x-axis 1st component of PCA
On y-axis 2nd component of PCA

(A) Violin plot of PC1 values



(B) On x-axis Class of Cancer Types.
On y-axis scaling of PC1 values.

(A) Violin plot of PC2 values



(B) On x-axis Class of Cancer Types.
On y-axis scaling of PC2 values.

(C) Observation: We cannot apply PCA on categorical variables. We have to convert the categorical data into binary data to apply PCA on the dataset. After converting the categorical data into binary data we have to concat that converted binary data to the main dataset on which the PCA algorithm was implemented.

(D) Conclusion: After applying PCA on the lncRNA_5_Cancer's dataset, we can reduce the dataset of dimension 2529x12309 into just 2529x2 dimensions. First, we must do the feature scaling using standard scaler class form sklearn. For that we must separate the independent and dependent values. On the independent values we applied feature scaling. After applying, we use principal component analysis on the independent features.

T-Distributed Stochastic Neighbor Embedding (t-SNE)

(a) What it does:

T-Distributed Stochastic Neighbor Embedding (t-SNE) is a nonlinear dimensionality reduction technique that is particularly suitable for the visualization of high-dimensional datasets. It measures a similarity pair of examples in multi, high and also in the low dimensional space.

(b) How it does:

The algorithms start by computing the probability of similarity of points in a high-dimensional space and computing the probability of similarity of points in the corresponding low-dimensional space. Point similarity is calculated as the conditional probability that point A would choose point B as its neighbor if the neighbors were chosen in proportion to their probability density under a Gaussian (normal distribution) centered at A.

It then tries to minimize the difference between these conditional probabilities (or similarities) in the higher and lower dimensional space for a perfect representation of the data points in the lower dimensional space.

To measure the minimization of the sum of the difference of the conditional probability, t-SNE minimizes the sum of the Kullback-Leibler divergence of the total data points using the gradient descent method.

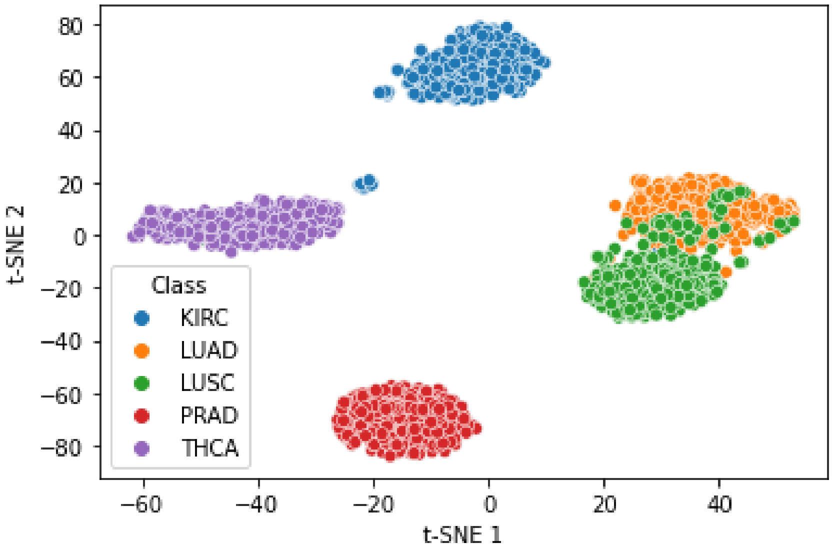
The Kullback-Leibler divergence, or KL divergence, is a measure of how one probability distribution differs from another, the expected probability distribution. Those interested in detailed information on how the algorithm works can refer to this research paper. More simply, t-distributed stochastic neighbor embedding (t-SNE) minimizes the divergence between two distributions: the distribution that measures the pairwise similarities of the input objects and the distribution that measures the pairwise similarities of the corresponding low-dimensional points in the embedding. In this way, t-SNE maps multidimensional data into a lower dimensional space and attempts to find patterns in the data by identifying observed clusters based on the similarity of multi-feature data points. However, after this process, the input elements are no longer identifiable, and no conclusions can be drawn based on the t-SNE output alone. So, it is primarily a technique of data exploration and visualization.

(c) Application:

It is widely used in image processing, NLP, genomic data, speech processing and also in cyber security and bioinformatics.

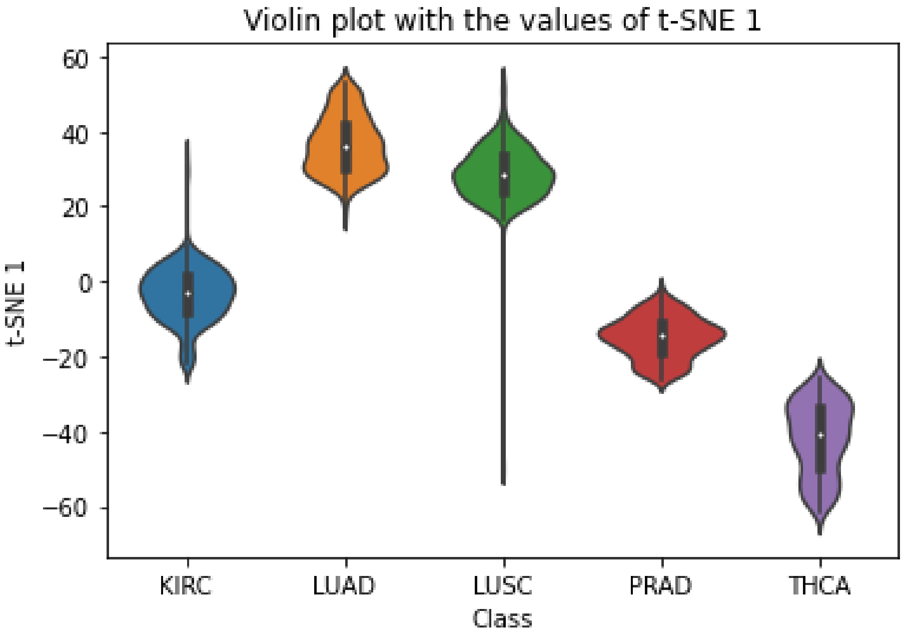
RESULTS for t-SNE

(A) t-SNE Visualization



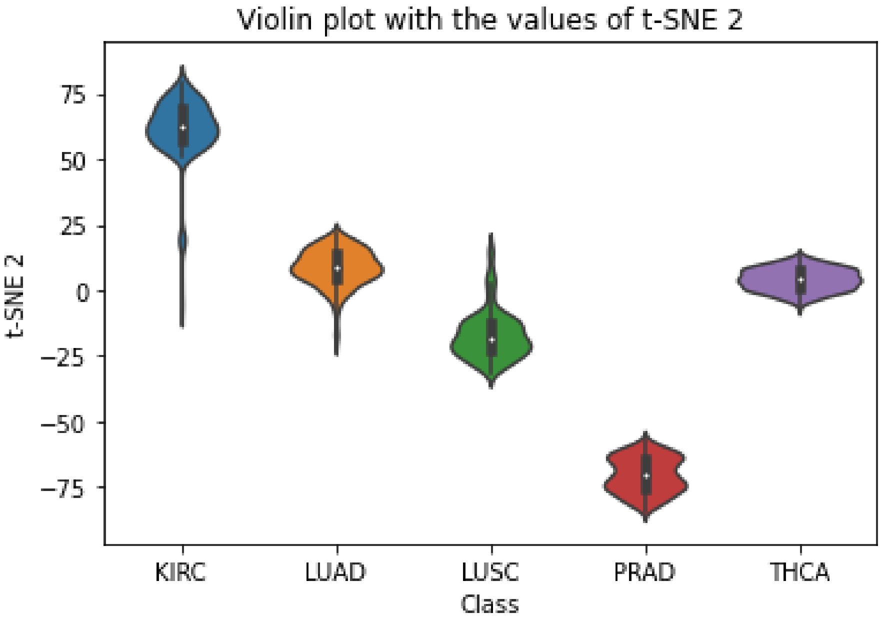
(B) On x-axis 1st component of t-SNE
On y-axis 2nd component of t-SNE

(A) Violin plot of t-SNE-1



(B) On x-axis Class of Cancer Types.
On y-axis scaling of 1st component of t-SNE.

(A) Violin plot of t-SNE-2



(B) On x-axis Class of Cancer Types.
On y-axis scaling of 2nd component of t-SNE.

(C) Observation:

The main primacy of t-SNE is that, it has the capability to hold the main pattern or structure and clusters the nearby points to one another in the giant dimensional data set in the chart. It delivers more insightful outcomes i.e., charts compared to other models or tools. It is also handy and easily adaptable. We need to be conscious and notice the output according to the parameters related to this model.

(D) Conclusion:

The main difference between the PCA and t-SNE was t-SNE concerns about the hyper parameters which includes perplexity, learning rate and total steps or iterations. It maintains using hyper parameters where as in PCA it maintains by the method of variance. We can say that the learning rate parameter is the complex parameter and decides how the model or algorithm behaves. We cannot or directly make the conclusions based on only the final output of the model, it will be mostly limits their usage to data exploration and visualization.