Name: Alif Moinul Alam

Panther ID: 6370828

Grade Student

Task 1 **Reduce the data dimension from 12,309 to two (PC1 and PC2) dimension.**

a) What it is or What it does: PCA: PCA, or Principal component analysis, the is the main linear algorithm for dimension reduction in unsupervised learning. This algorithm distinguishes and discards features that are less useful to make a valid approximation on a dataset.

b) How it does:

The Principal Components have now got nothing to do with the original features.

Step 1: Get your data: Load the data from the data set

Step 2: Give your data a structure.

Preprocess the data set to filter out the unnecessary data. Separating out the features

Separating out the target which is class.

Step 3: Import PCA from Sklearn and set the number of components we want

Step 4: Fit the data with a formula and put it into transform

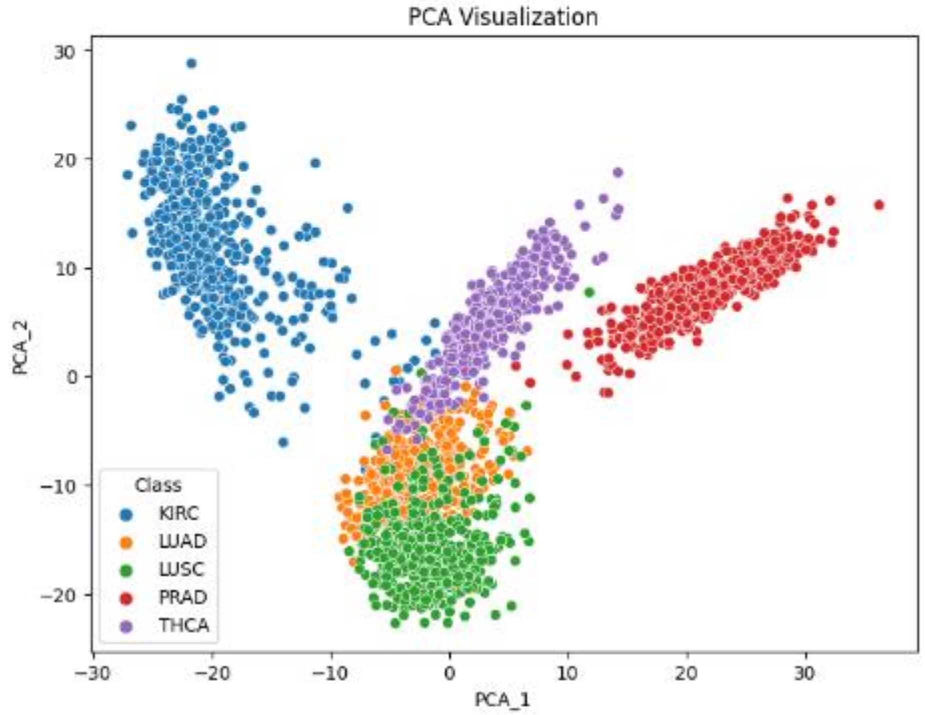
Step 5: Visualize the Data with Scatter plot.

c) Some of the applications of Principal Component Analysis (PCA) are:

Spike-triggered covariance analysis in Neuroscience

- Quantitative Finance
- Image Compression
- Facial Recognition
- Other applications like Medical Data correlation

Describe results: (a)



(b) Describe the figure and table: Analysis the data after reduce the dimension there PCA 2 in the Y axes and PCA 1 in the X axes.

The discussion of the result I got 5 output from the data and it is supervised learning and it's output are levels from Class columns. The categories are ['KIRC','LUAD','LUSC','PRAD','THCA'].

c) Your observation about the figure and table: The principal application of PCA is dimension reduction. If you have high dimensional data, PCA allows you to reduce the dimensionality of your data so the bulk of the variation that exists in your data across many high dimensions is captured in fewer dimensions. PCA is used abundantly in all forms of analysis - from Neuroscience to Quantitative Finance. PCA has wide-spread applications in various industries.

d) Conclusion: Principal component analysis, or PCA, is a statistical procedure that allows you to summarize the information content in large data tables by means of a smaller set of "summary indices" that can be more easily visualized and analyzed.

Task 2: Draw two violin plots – one with the values of PC1 and the other with PC2.

a) A violint plot allow to visualize the distribution of a numeric variable for one or several groups. In a violin plot, individual density curves are built around center lines, rather than stacked on baselines. Other than this difference in display pattern, curves in a violin plot follow the exact same construction and interpretation.

b) How its work:

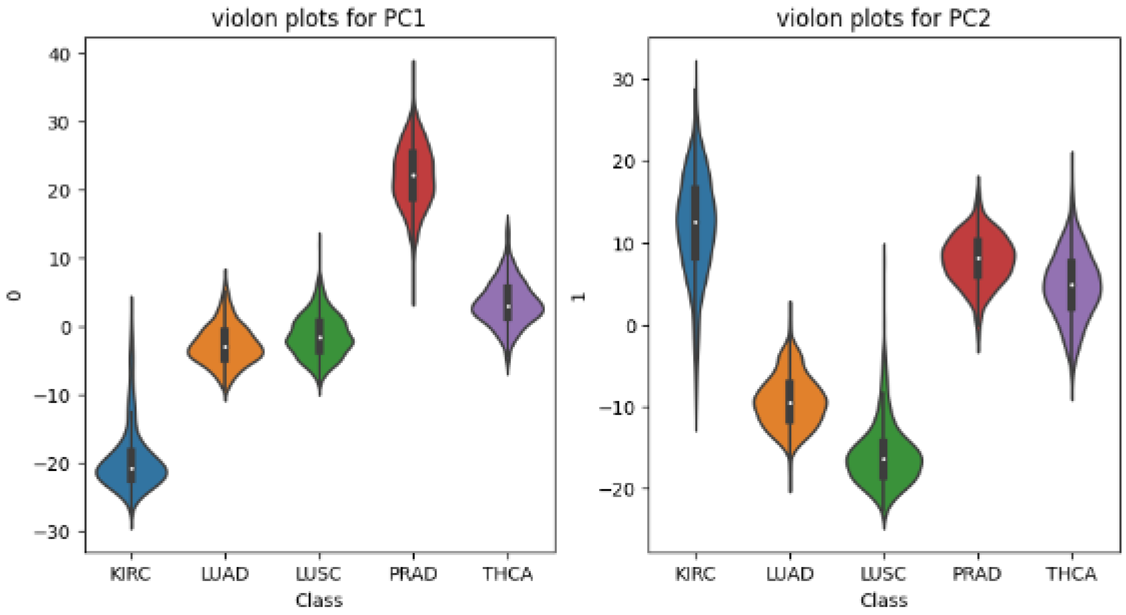
   Step 01: After reduce the dimension from the large
   Step 02: Import the seaborn
   Step 03: plot it with PC1 and PC2

Step 04: Show the result

C) Application of a Violin plot : visualize the distribution of numerical data.

(a) Put Figure/Table number and Title: On top of the table, and

bottom of the figure.



(b) Describe the figure and table. The categories are ['KIRC','LUAD','LUSC','PRAD','THCA']. With 1st it shows with PC1 and 2nd it shows with PC2. In the X axis the name of the categories.

(c) Your observation: Violin plots are used when I want to observe the distribution of numeric data, and are especially useful when I want to make a comparison of distributions between multiple groups.

(d) Conclusion: Violin plots are less common than other plots like the box plot due to the additional complexity of setting up the kernel and bandwidth. They can also be visually noisy, especially with an overlaid chart type. If you are trying to think of a chart to demonstrate findings to an audience unfamiliar with the violin plot, it might be better to go with a simpler and more straightforward visualization like the box plot.

Task 3: Repeat task 1 using t-SNE library. Plot the data in reduced dimension using

two t-SNE components (t-SNE 1 and t-SNE 2).

a) t-SNE: T-distributed stochastic neighbor embedding (t-SNE) is a statistical method for visualizing high-dimensional data.(t-SNE) t-Distributed Stochastic Neighbor Embedding is a non-linear dimensionality reduction algorithm used for exploring high-dimensional data. It maps multi-dimensional data to two or more dimensions suitable for human observation. With help of the t-SNE algorithms, you may have to plot fewer exploratory data analysis plots next time you work with high dimensional data.

b) How it works

Step 1

To run t-SNE in Python, we will use the given dataset.I have also used scRNA-seq data for t-SNE visualization

Step 2: Give your data a structure.

Preprocess the data set to filter out the unnecessary data. Separating out the features

Separating out the target which is class.

Step 3: Import t-SNE from Sklearn and set the number of components we want

Step 4: Fit the data with a formula and put it into transform

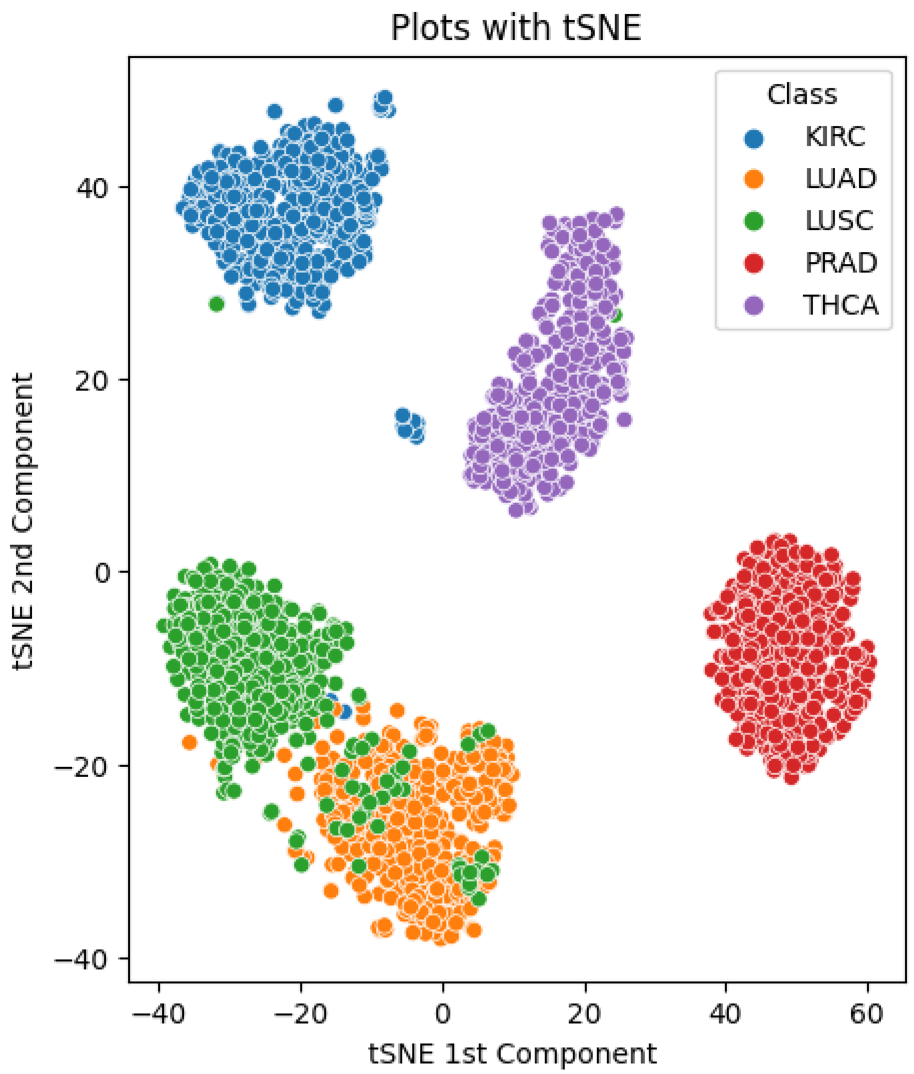
Step 5: Visualize the Data with Scatter plot.

c)Applications of t-SNE:

t-SNE has been used for visualization in a wide range of applications, including genomics, computer security research, natural language processing, music analysis, cancer research, bioinformatics, geological domain interpretation, and biomedical signal processing.

Describe the picture

a)Put Figure/Table number and Title: On top of the table, and



(b) Describe the figure and table:

In the X axis it shows t-SNE 1st Component and y axis it shows t SNE 2nd Component. And there are 5 cluster in the image.

(c) Your observation about the figure and table: In the question its told to apply t-SNE in the data set and get the division. In here There are five groups from the classes. These are['KIRC','LUAD','LUSC','PRAD','THCA']. And separate groups

(d) Conclusion:

t-SNE is a great tool to understand high-dimensional datasets. It might be less useful when you want to perform dimensionality reduction for ML training (cannot be reapplied in the same way). It's not deterministic and iterative so each time it runs, it could produce a different result. But even with that disadvantages it still remains one of the most popular method in the field.

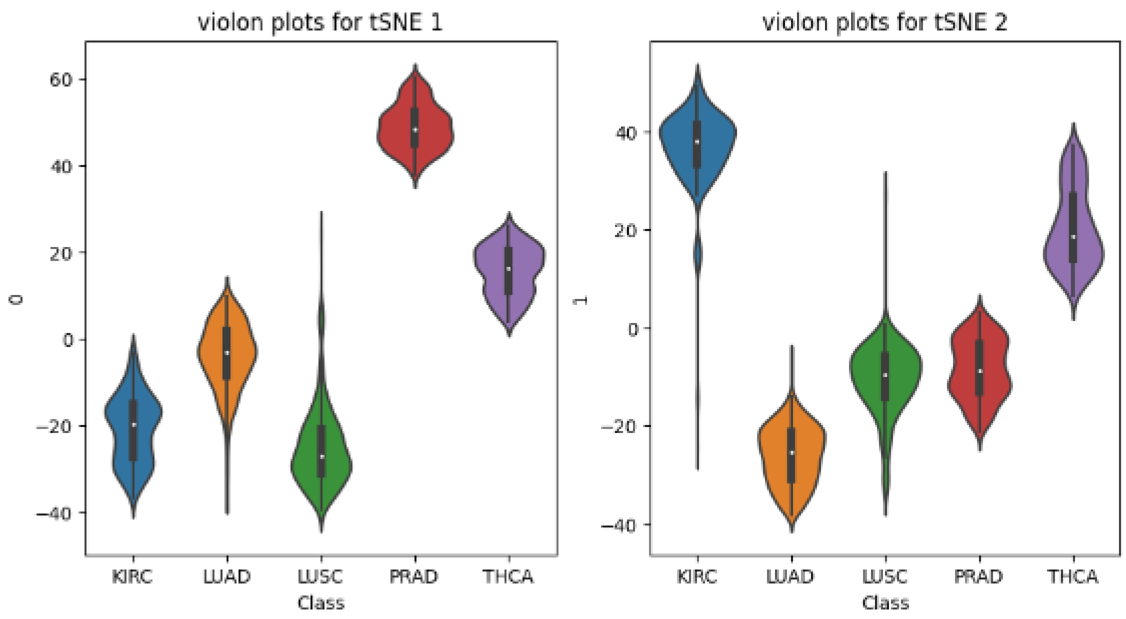
Task 4: Draw two violin plots – one with the values of t-SNE 1 and the other with tSNE 2.

a) Violin plots: A violin plot allow to visualize the distribution of a numeric variable for one or several groups. In a violin plot, individual density curves are built around center lines, rather than stacked on baselines. Other than this difference in display pattern, curves in a violin plot follow the exact same construction and interpretation.

b) How its work:
Step 01: After reduce the dimension from the large
Step 02: Import the seaborn
Step 03: plot it with Tsne-1 and Tsne-2
Step 04: Show the result

c) Application: Visualize the distribution of numerical data.

a)Put Figure/Table number and Title: On top of the table, and



(b) Describe the figure and table. The categories are ['KIRC','LUAD','LUSC','PRAD','THCA']. With 1st it shows with tSNE1 and 2nd it shows with tSNE2. In the X axis the name of the categories.

(c) Your observation: Violin plots are used when I want to observe the distribution of numeric data, and are especially useful when I want to make a comparison of distributions between multiple groups.

(d) Conclusion: Violin plots are less common than other plots like the box plot due to the additional complexity of setting up the kernel and bandwidth. They can also be visually noisy, especially with an overlaid chart type. If you are trying to think of a chart to demonstrate findings to an audience unfamiliar with the violin plot, it might be better to go with a simpler and more straightforward visualization like the box plot.