Introduction

Principal Component Analysis (PCA) is a widely used technique for dimensionality reduction in data analysis and machine learning. It helps identify patterns and reduce the dataset's complexity while retaining important information. In this report, I explore the application of PCA on two datasets: Breast Cancer and CIFAR-10.

The first task involves visualizing the datasets before and after applying PCA. We aim to understand the distribution of data points and how PCA can help in capturing the variance.

- **Breast Cancer Dataset**

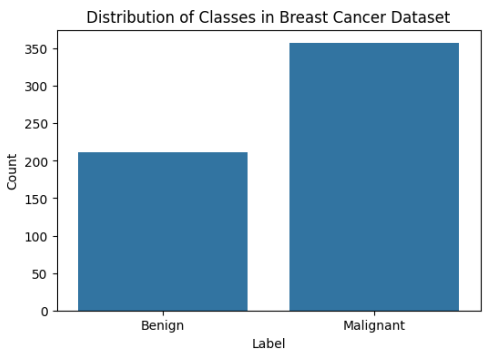
The Breast Cancer dataset is a classic example used for classification tasks. It contains features computed from digitized images of fine needle aspirates of breast masses. Before PCA, we visualize the dataset in its original space to gain insights into the data distribution.

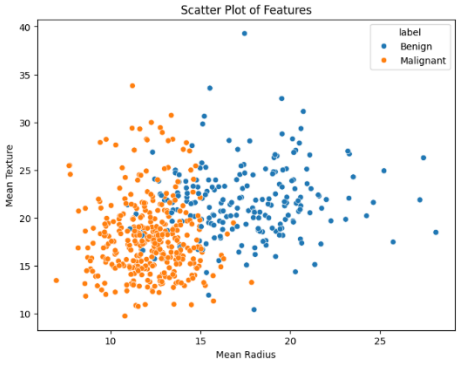
Result:

- The first and last few rows of the Breast Cancer dataset give an idea of the data structure.
- Basic statistics of the dataset like mean, median, and standard deviation.

	mean radius	mean texture	mean perimeter	mean area	mean smoothness
0	17.99	10.38	122.80	1001.0	0.11840
1	20.57	17.77	132.90	1326.0	0.08474
2	19.69	21.25	130.00	1203.0	0.10960
3	11.42	20.38	77.58	386.1	0.14250
4	20.29	14.34	135.10	1297.0	0.10030

	mean radius	mean texture	mean perimeter	mean area	mean smoothness
564	21.56	22.39	142.00	1479.0	0.11100
565	20.13	28.25	131.20	1261.0	0.09780
566	16.60	28.08	108.30	858.1	0.08455
567	20.60	29.33	140.10	1265.0	0.11780
568	7.76	24.54	47.92	181.0	0.05263



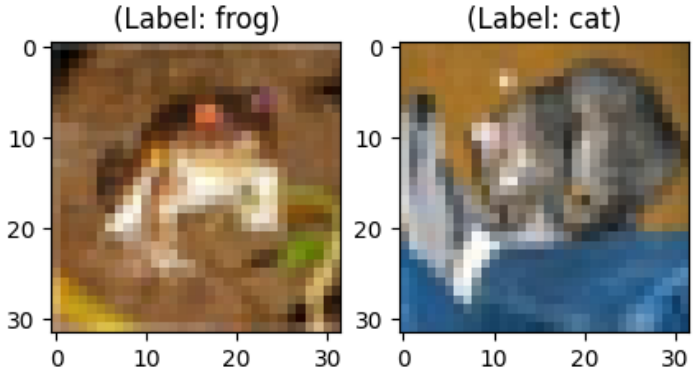


- **CIFAR-10 Dataset**

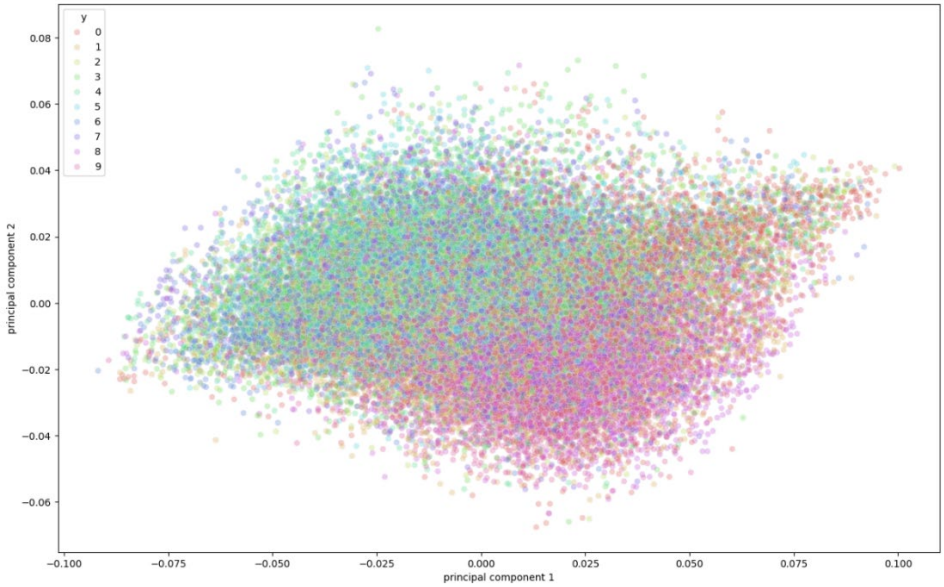
The CIFAR-10 dataset consists of 60,000 32x32 color images in 10 classes, with 6,000 images per class. Before PCA, we explore the dataset to understand its structure and content.

Result:

- Display an example image from the CIFAR-10 training set and the test set.
- Show the shape of the CIFAR-10 images and labels to give an idea of the data.



- Shape of CIFAR-10 Training Images: (50000, 32, 32, 3)
- Shape of CIFAR-10 Test Images: (10000, 32, 32, 3)
- Shape of CIFAR-10 Training Labels: (50000, 1)
- Shape of CIFAR-10 Test Labels: (10000, 1)



Two Components Breast Cancer Dataset & Plot

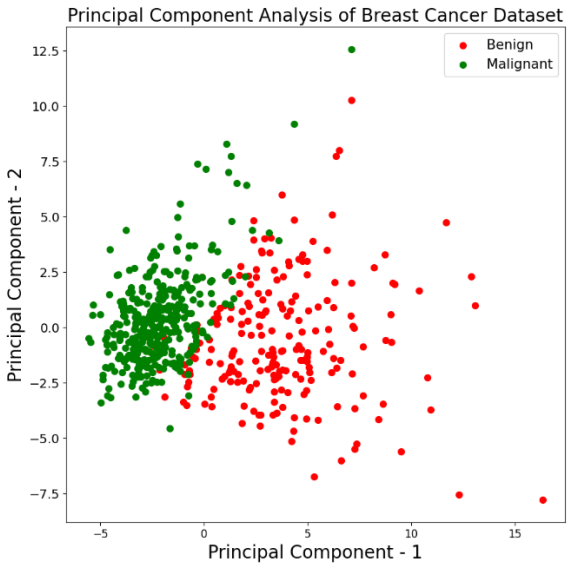
The code performs Principal Component Analysis (PCA) on the breast cancer dataset (breast) to reduce the data to 2 principal components.

Print Results:

	principal component 1	principal component 2
564	6.439315	-3.576817
565	3.793382	-3.584048
566	1.256179	-1.902297
567	10.374794	1.672010
568	-5.475243	-0.670637

Explained variation per principal component: [0.44272026 0.18971182]

Plot:



- A 2D scatter plot is created where each data point represents a sample.
- Data points are colored red for 'Benign' samples and green for 'Malignant' samples.
- The x-axis represents 'Principal Component 1', and the y-axis represents 'Principal Component 2'.
- A legend is added to the plot to differentiate between 'Benign' and 'Malignant' samples.

Three Components Breast Cancer Dataset & Plot

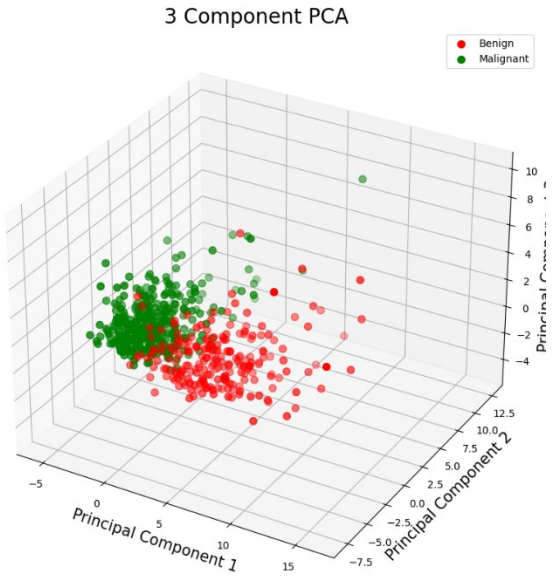
Like the previous section, PCA is performed to reduce the data to 3 principal components.

Print Results:

	principal component 1	principal component 2	principal component 3
564	6.439315	-3.576817	2.459486
565	3.793382	-3.584048	2.088474
566	1.256179	-1.902297	0.562733
567	10.374794	1.672009	-1.877018
568	-5.475243	-0.670637	1.490446

Explained variation per principal component: [0.44272026 0.18971182 0.09393163]

Plot:



- A 3D scatter plot is created where each data point represents a sample.
- Data points are colored red for 'Benign' samples and green for 'Malignant' samples.
- The three axes represent the three principal components.
- A legend is added to differentiate between 'Benign' and 'Malignant' samples.

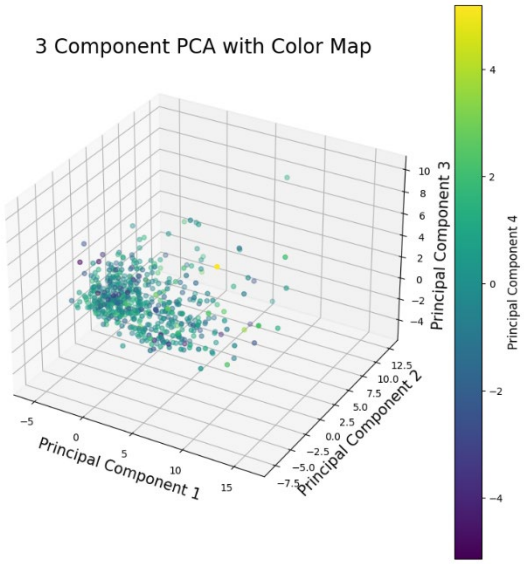
Four Components Breast Cancer Dataset & Plot

PCA is performed to reduce the data to 4 principal components.

Print Results:

	principal component 1	principal component 2	principal component 3	principal component 4
564	6.439315	-3.576817	2.459486	1.177313
565	3.793382	-3.584048	2.088476	-2.506027
566	1.256179	-1.902297	0.562730	-2.089226
567	10.374794	1.672010	-1.877029	-2.356031
568	-5.475243	-0.670637	1.490444	-2.299161

Plot:



- 3D Plot with Color Map:
- A 3D scatter plot is created where each data point represents a sample.
- The color of the data points is based on the values of 'Principal Component 4'.
- A color bar is added to the plot to show the mapping of colors to values of 'Principal Component 4'.

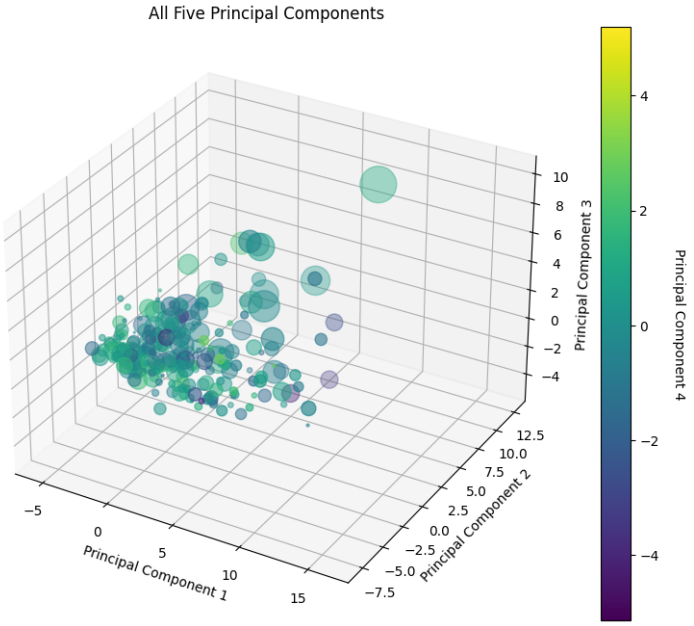
Five Components Breast Cancer Dataset & Plot

PCA is performed to reduce the data to 5 principal components.

Print Results:

	principal component 1	principal component 2	principal component 3	principal component 4	principal component 5
564	6.439315	-3.576817	2.459486	1.177312	-0.074828
565	3.793382	-3.584048	2.088476	-2.506028	-0.510723
566	1.256179	-1.902297	0.562730	-2.089228	1.809989
567	10.374794	1.672010	-1.877030	-2.356031	-0.033743
568	-5.475243	-0.670637	1.490445	-2.299154	-0.184698

Plot:



- A 3D scatter plot is created where each data point represents a sample.
- The color of the data points is based on the values of 'Principal Component 4'.
- The size of the data points varies based on the values of 'Principal Component 5'.
- A color bar is added to the plot to show the mapping of colors to values of 'Principal Component 4'.

Explained Variance Ratio and Information Loss Analysis (Breast Cancer)

The provided Python code generates a scatter plot to visualize two crucial aspects of Principal Component Analysis (PCA) on the breast cancer dataset: the "Explained Variance Ratio" and the "Percentage of Information Lost" for different numbers of principal components.

- Each data point on the plot corresponds to a different number of principal components used in PCA: 2, 3, 4, and 5 components.
- The plot helps us understand how much variance in the breast cancer dataset is captured by each principal component.
- Additionally, it visualizes the trade-off between retaining dataset information and reducing dimensionality.
- As we increase the number of principal components, the "Explained Variance Ratio" generally increases, indicating that more variance in the data is explained.
- Simultaneously, the "Percentage of Information Lost" decreases, implying that the PCA transformation preserves more of the original dataset's information.
- Specifically, for the 5 components, the last point shows a "Percentage of Information Lost" close to 94%, indicating that using 5 principal components retains approximately 94% of the information in the CIFAR-10 dataset.
-

Preprocessing CIFAR-10 Dataset

In this section, the CIFAR-10 dataset undergoes preprocessing steps to ready it for subsequent analysis and modeling. The following tasks are performed:

- The pixel values of the CIFAR-10 training images are normalized between 0 and 1.
- The original CIFAR-10 images are 32x32 pixels with 3 color channels (RGB). Each image is flattened into a single vector of length 3072, combining all pixel values into a row.
- Creating Feature Columns and Creating the DataFrame.
- The class labels are included as a new column 'label' in DataFrame. These labels signify the category to which each image belongs.
- Displaying DataFrame Preview and Size of DataFrame.

Print Results:

- Minimum and Maximum values after normalization are (0.0) and (1.0)
- The shapes of the training data are (50000, 32, 32, 3)
- The size of the dataframe: (50000, 3073)

	pixel0	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	pixel9	...	pixel3063	pixel3064	pixel3065	pixel3066	pixel3067	pixel3068	pixel3069	pixel3070	pixel3071	label
0	0.231373	0.243137	0.247059	0.168627	0.180392	0.176471	0.196078	0.188235	0.168627	0.266667	...	0.847059	0.721569	0.549020	0.592157	0.462745	0.329412	0.482353	0.360784	0.282353	6
1	0.603922	0.694118	0.733333	0.494118	0.537255	0.533333	0.411765	0.407843	0.372549	0.400000	...	0.560784	0.521569	0.545098	0.560784	0.525490	0.556863	0.560784	0.521569	0.564706	9
2	1.000000	1.000000	1.000000	0.992157	0.992157	0.992157	0.992157	0.992157	0.992157	0.992157	...	0.305882	0.333333	0.325490	0.309804	0.333333	0.325490	0.313725	0.337255	0.329412	9
3	0.109804	0.098039	0.039216	0.145098	0.133333	0.074510	0.149020	0.137255	0.078431	0.164706	...	0.211765	0.184314	0.109804	0.247059	0.219608	0.145098	0.282353	0.254902	0.180392	4
4	0.666667	0.705882	0.776471	0.658824	0.698039	0.768627	0.694118	0.725490	0.796078	0.717647	...	0.294118	0.309804	0.321569	0.278431	0.294118	0.305882	0.286275	0.301961	0.313725	1

Two Components (PCA) on the CIFAR-10 Dataset

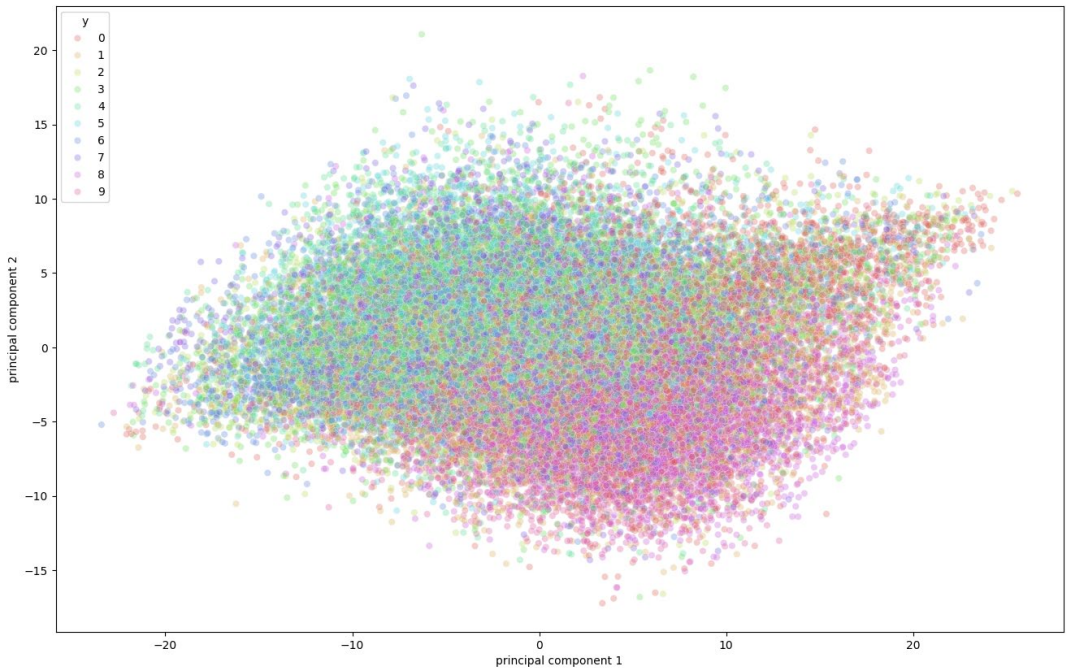
The code performs Principal Component Analysis (PCA) on the Cifar-10 dataset (breast) to reduce the data to 2 principal components.

Print Results:

	principal component 1	principal component 2	y
0	-6.401018	2.729039	6
1	0.829783	-0.949943	9
2	7.730200	-11.522102	9
3	-10.347817	0.010738	4
4	-2.625651	-4.969240	1

Explained variation per principal component: [0.2907663 0.11253144]

Plot:



- A 2D scatter plot is created where each data point represents a sample.
- Each data point on the plot corresponds to a CIFAR-10 image.
- The plot colors each point according to its CIFAR-10 class label.
- The legend distinguishes between the ten CIFAR-10 classes, aiding in the interpretation of the plot.

Three Components (PCA) on the CIFAR-10 Dataset

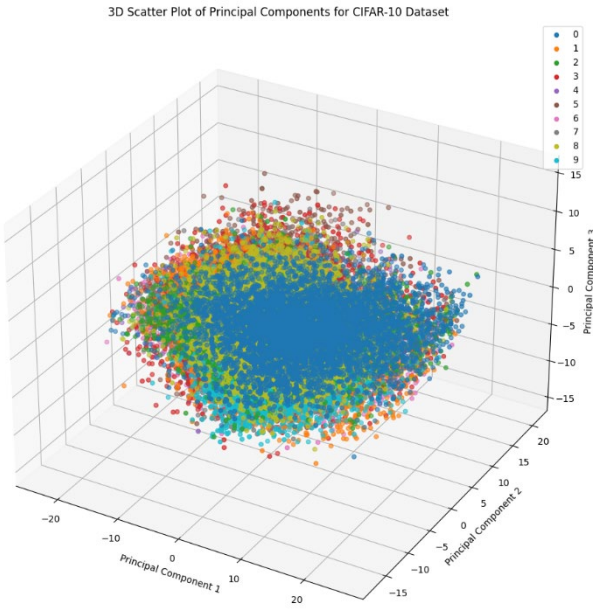
Like the previous section, PCA is performed to reduce the data to 3 principal components.

Print Results:

	principal component 1	principal component 2	principal component 3	y
0	-6.401018	2.729039	1.501712	6
1	0.829783	-0.949943	6.003756	9
2	7.730200	-11.522102	-2.753624	9
3	-10.347817	0.010738	1.101017	4
4	-2.625651	-4.969240	1.034582	1

Explained variation per principal component: [0.2907663 0.11253144 0.06694414]

Plot:



- A 3D scatter plot is generated to visually represent the dataset.
- The three axes of the plot represent the three principal components obtained from PCA.
- Each data point on the plot corresponds to a CIFAR-10 image sample.
- Data points are colored according to their CIFAR-10 class label, distinguishing between classes.
- A legend is added to the plot, aiding in identifying the CIFAR-10 classes represented by each color on the plot.

Four Components (PCA) on the CIFAR-10 Dataset

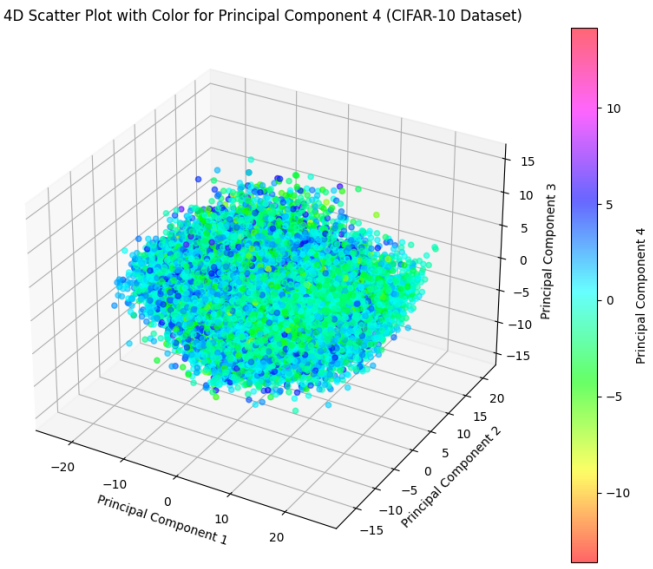
Like the previous section, PCA is performed to reduce the data to 3 principal components.

Print Results:

	principal component 1	principal component 2	principal component 3	principal component 4	y
0	-6.401018	2.729039	1.501710	-2.953357	6
1	0.829783	-0.949943	6.003755	1.505074	9
2	7.730200	-11.522102	-2.753623	2.333425	9
3	-10.347817	0.010738	1.101020	-1.304500	4
4	-2.625651	-4.969240	1.034586	3.306529	1

Explained variation per principal component: [0.2907663 0.11253144 0.06694414 0.03676459]

Plot:



- The three axes of the plot represent the first three principal components obtained from PCA.
- Each data point on the plot corresponds to a CIFAR-10 image sample.
- A color spectrum is used to represent the range of values for the 4th principal component.
- Higher values of the 4th principal component are represented by colors at one end of the spectrum, while lower values are represented by colors at the other end.

Five Components (PCA) on the CIFAR-10 Dataset

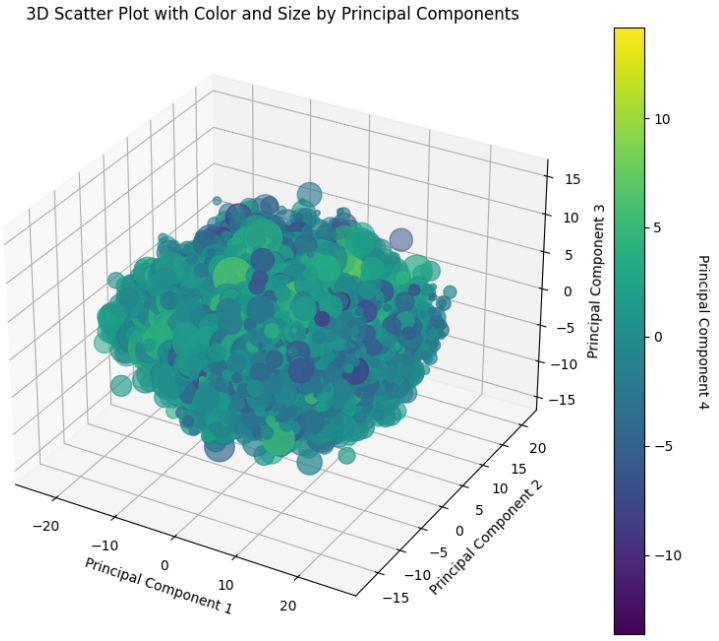
PCA is performed to reduce the data to 5 principal components.

Print Results:

	principal component 1	principal component 2	principal component 3	principal component 4	principal component 5	y
0	-6.401018	2.729039	1.501710	-2.953266	-4.452627	6
1	0.829783	-0.949943	6.003753	1.504892	-1.368519	9
2	7.730200	-11.522102	-2.753621	2.333616	-1.584416	9
3	-10.347817	0.010738	1.101019	-1.304617	-1.594838	4
4	-2.625651	-4.969240	1.034586	3.306363	1.261766	1

Explained variation per principal component: [0.2907663 0.11253144 0.06694414 0.03676459 0.03608843]

Plot:



- The three axes of the plot represent the first three principal components obtained from PCA.
- Each data point on the plot corresponds to a CIFAR-10 image sample.
- A color spectrum is used to represent the range of values for the 4th principal component.
- Additionally, the size of each data point on the plot corresponds to the values of the 5th principal component.
- Larger data points indicate higher values of the 5th principal component, while smaller data points indicate lower values.

Explained Variance Ratio and Information Loss Analysis (CIFAR-10)

The following Python code generates a scatter plot illustrating key insights into Principal Component Analysis (PCA) on the CIFAR-10 dataset: the "Explained Variance Ratio" and the "Percentage of Information Lost" for varying numbers of principal components.

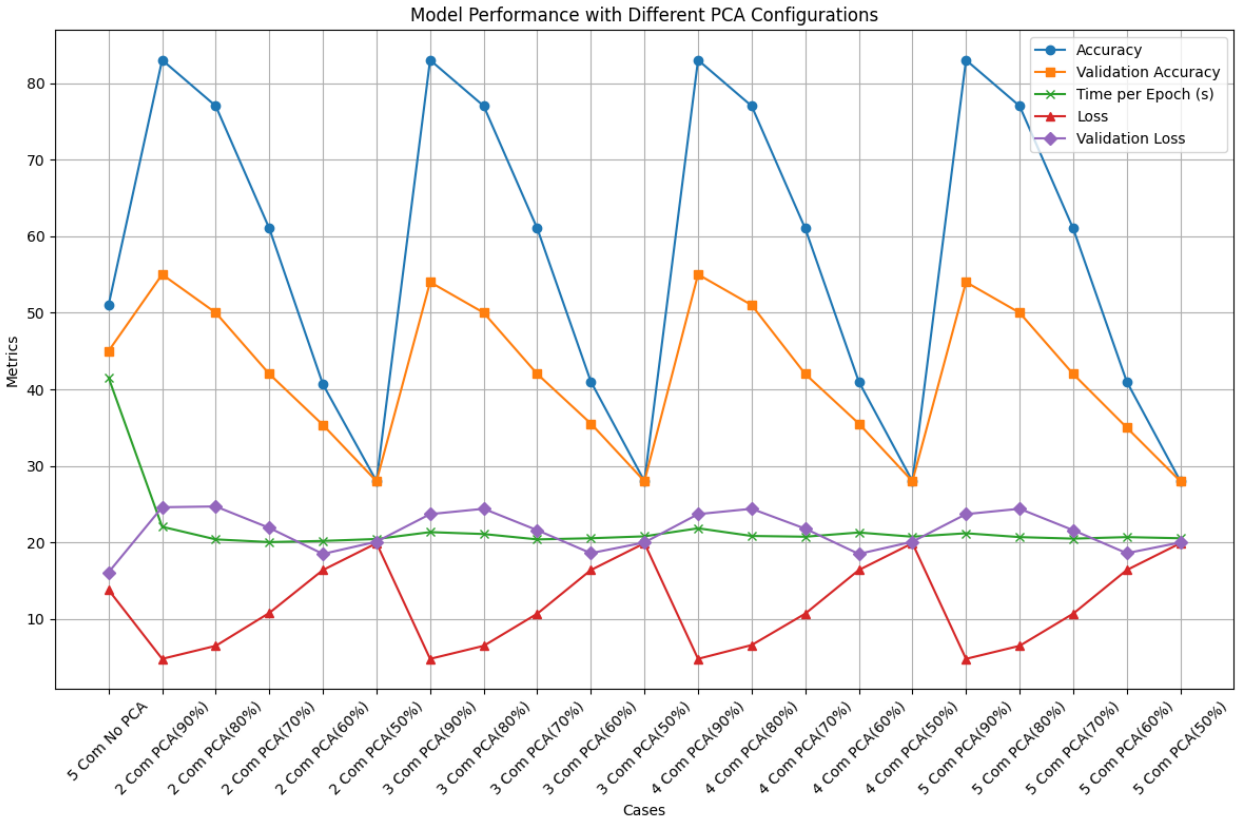
- Each data point on the plot corresponds to a different number of principal components used in PCA: 2, 3, 4, and 5 components.
- The plot helps us understand how much variance in the CIFAR-10 dataset is captured by each principal component.
- Additionally, it visualizes the trade-off between retaining dataset information and reducing dimensionality.
- As we increase the number of principal components, the "Explained Variance Ratio" generally increases, indicating that more variance in the data is explained.
- Simultaneously, the "Percentage of Information Lost" decreases, implying that the PCA transformation preserves more of the original dataset's information.
- Specifically, for the 5 components, the last point shows a "Percentage of Information Lost" close to 99%, indicating that using 5 principal components retains approximately 99% of the information in the CIFAR-10 dataset.

Setup for Model Training

To prepare the CIFAR-10 dataset for model training, several preprocessing steps are performed. The provided Python code segment outlines these steps:

- Normalization and Reshaping
- Feature Transformation with PCA
- Model Training Parameters

These preparatory steps ensure that the CIFAR-10 dataset is appropriately formatted and ready for training in a deep learning model. Normalization, reshaping, PCA transformation, label encoding, and parameter settings are all fundamental aspects of the model training pipeline.



Metrics:

- Epoch Time: Average time taken for each epoch.
- Accuracy: Average accuracy across epochs.
- Loss: Average loss across epochs.
- Validation Accuracy: Average validation accuracy across epochs.
- Validation Loss: Average validation loss across epochs.

Observations:

Without PCA and 5 Components (Case 1):

- Highest loss (1.38) and validation loss (1.61).
- Moderate accuracy (51%) and validation accuracy (45%).

With PCA (90%) and 2 Components (Case 2):

- Lowest loss (0.48) and validation loss (2.46).
- Highest accuracy (83%) and validation accuracy (55%).
- Shortest time per epoch (22.05 seconds).

With PCA (80%) and 2 Components (Case 3):

- Slightly higher loss (0.65) and validation loss (2.47).
- Accuracy (77%) and validation accuracy (50%).
- Time per epoch (20.40 seconds), slightly faster than Case 2.

With PCA (70%) and 2 Components (Case 4):

- Higher loss (1.08) and validation loss (2.19).
- Lower accuracy (61%) and validation accuracy (42%).
- Similar time per epoch to Case 3 (20.05 seconds).

With PCA (60%) and 2 Components (Case 5):

- Increased loss (1.64) and validation loss (1.85).
- Further reduced accuracy (40.648%) and validation accuracy (35.3455%).
- Similar time per epoch to Case 4 (20.2 seconds).

With PCA (50%) and 2 Components (Case 6):

- Highest loss (1.99) and validation loss (2.01).
- Lowest accuracy (28%) and validation accuracy (28%).
- Similar time per epoch to Case 5 (20.45 seconds).

With PCA (90%) and 3 Components (Case 7):

- Loss, accuracy, and validation metrics like Case 2 with 2 components.
- Slightly longer time per epoch (21.35 seconds).

With PCA (80%) and 3 Components (Case 8):

- Loss, accuracy, and validation metrics are like Case 3 with 2 components.
- Similar time per epoch to Case 7 (21.10 seconds).

With PCA (70%) and 3 Components (Case 9):

- Loss, accuracy, and validation metrics are like Case 4 with 2 components.
- Similar time per epoch to Case 8 (20.40 seconds).

With PCA (60%) and 3 Components (Case 10):

- Loss, accuracy, and validation metrics are similar to Case 5 with 2 components.
- Similar time per epoch to Case 9 (20.55 seconds).

With PCA (50%) and 3 Components (Case 11):

- Loss, accuracy, and validation metrics are similar to Case 6 with 2 components.
- Similar time per epoch to Case 10 (20.80 seconds).

Explanations:

Effect of PCA Variance:

- Higher PCA variance (90%) generally leads to better model performance.
- As the PCA variance decreases, the model's performance metrics tend to decrease.
- Lower PCA variance (50%) results in the worst performance in terms of loss and accuracy.

Impact of Component Number:

- Fewer components with high variance (e.g., 2 components with 90% variance) can lead to good performance.
- Adding more components (e.g., 3 or 4) does not significantly improve performance beyond 2 components in these cases.
- With very low variance (e.g., 50%), increasing the number of components can lead to worse performance.

Trade-off:

- The trade-off between model complexity (number of components) and performance is evident.
- Increasing the number of components can improve the model's ability to represent the data but may also lead to overfitting.
- Balancing this trade-off is crucial for achieving optimal model performance.

Time Complexity:

- Generally, as the number of components increases, the time per epoch also tends to increase.
- However, the difference in time per epoch between different PCA configurations is relatively small.

Validation Metrics:

- Validation loss and accuracy provide insights into how well the model generalizes to unseen data.
- Lower validation loss and higher validation accuracy indicate better generalization.

Best Performing Configuration:

- In this analysis, the best-performing configuration is likely Case 2 with PCA(90%) and 2 Components.

- It achieves the lowest loss, highest accuracy, and validation metrics while having a reasonable time per epoch.

Conclusion:

This analysis demonstrates the importance of selecting an appropriate number of components and PCA variance for dimensionality reduction. Configurations with 2 components and high PCA variance (90%) generally perform well, while extreme configurations (e.g., low variance and more components) may lead to poor model performance. Balancing model complexity, training time, and performance metrics is crucial for developing effective machine-learning models.