"Utilizing Principal Component Analysis (PCA) for Data Visualization and Dimensionality Reduction"

Prakash Kumar
Department of Computer Science
Lovely Professional University,
Phagawara,Punjab,144411
prakash.12019155@lpu.in

Reg no : 12019155

Dr. Usha Mittal
Department of Computer Science
Lovely Professional University
Phagawara,Punjab,144411
Ushamittal123@gmail.com

Abstract*— Principal Component Analysis (PCA) stands as a fundamental unsupervised learning tool, most notably recognized for its prowess in dimensionality reduction. However, its utility transcends mere data preparation; it acts as a potent aid in visualizing intricate datasets, rendering complex information comprehensible. This paper serves as a guide, demystifying the process of employing PCA for data visualization. It delves into deciphering high-dimensional data, elucidates the concept of explained variance within PCA, and demonstrates how these visual insights contribute to determining optimal parameters for dimensionality reduction. Split into two distinct sections, the paper starts by unveiling the challenges of visualizing high-dimensional data through scatter plots. It showcases the limitations of traditional 2D and 3D plotting techniques, emphasizing the pivotal role PCA plays in reducing dimensions for enhanced data visualization. Utilizing real-world datasets like the wine dataset, this paper showcases the transformative power of PCA in rendering distinct class boundaries, enabling clear classification. It underscores the sensitivity of PCA to data scaling and its pivotal impact on visualization quality. The second segment navigates through the notion of explained variance within PCA. It highlights the step-by-step removal of principal components and its visible impact on the dataset's structure. By removing components and visualizing the consequences, it illustrates the proportional loss of information and its correlation with the explained variance ratios. Additionally, this report extends its exploration to the application of PCA in machine learning tasks. It exemplifies how PCA aids in retaining substantial dataset information, allowing efficient classification with fewer features. This practical implementation further underscores the efficacy of PCA in preserving data integrity. Conclusively, this paper equips the reader with a comprehensive understanding of PCA's pivotal role in data visualization and dimensionality reduction, offering a deeper comprehension of the underlying mechanisms and their practical applications in machine learning endeavors.

keywords— PCA (Principal Component Analysis),Unsupervised Learning,Dimensionality Reduction,Data Visualization,Explained Variance,High-Dimensional Data,Scatter Plots,Feature Selection,Model Optimization,Information Retention

1. INTRODUCTION

This article discusses the practical application of Principal Component Analysis (PCA), an unsupervised machine learning technique. PCA is renowned for its role in dimensionality reduction, but it also offers significant advantages in data visualization. By visualizing complex datasets, analysts can gain valuable insights and make informed decisions regarding machine learning models. This report provides insights into leveraging PCA for data visualization and guiding dimensionality reduction by observing explained variance. It equips readers with the knowledge to:

Visualize high-dimensional data effectively.

Understand the concept of explained variance in PCA.

Observe and interpret explained variance results obtained from PCA on high-dimensional datasets.

This tutorial is divided into two parts; they are:

- Scatter plot of high-dimensional data
- Visualizing the explained variance

I. Scatter plot of high dimensional data

Visualization is a crucial step to get insights from data. We can learn from the visualization whether a pattern can be observed and hence estimate which machine learning model is suitable.

It is easy to depict things in two dimension. Normally a scatter plot with x- and y-axis are in two dimensional. Depicting things in three dimensional is a bit challenging but not impossible. In matplotlib, for example, can plot in 3D. The only problem is on paper or on screen, we can only look at a 3D plot at one viewport or projection at a time. In matplotlib, this is controlled by the degree of elevation and azimuth. Depicting things in four or five dimensions is impossible

because we live in a three-dimensional world and have no idea of how things in such a high dimension would look like.

This is where a dimensionality reduction technique such as PCA comes into play. We can reduce the dimension to two or three so we can visualize it. Let's start with an example.

Dataset

We start with the wine dataset, which is a classification dataset with 13 features (i.e., the dataset is 13 dimensional) and 3 classes. There are 178 samples:

Data Set Characteristics:

Number of Instances:

178

Number of Attributes:

13 numeric, predictive attributes and the class

Attribute Information:

- Alcohol
- Malic acid
- Ash
- Alcalinity of ash
- Magnesium
- Total phenols
- Flavanoids
- Nonflavanoid phenols
- Proanthocyanins
- Color intensity
- Hue
- OD280/OD315 of diluted wines
- Proline

Data - Preprocessing

Among the 13 features, we can pick any two and plot with matplotlib (we color-coded the different classes using the c argument)

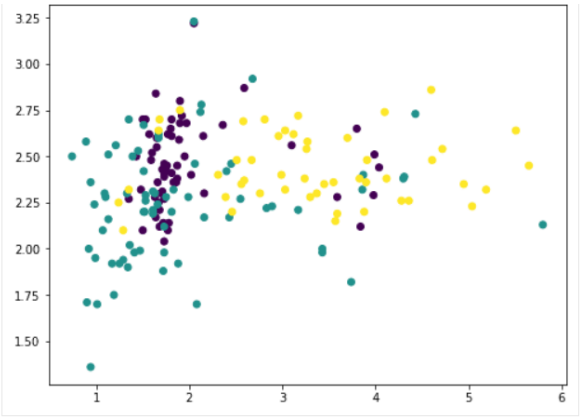

Fig. 1. Any two Among the 13 features and plot with matplotlib (color-coded the different classes using the c argument)

or we can also pick any three and show in 3D:

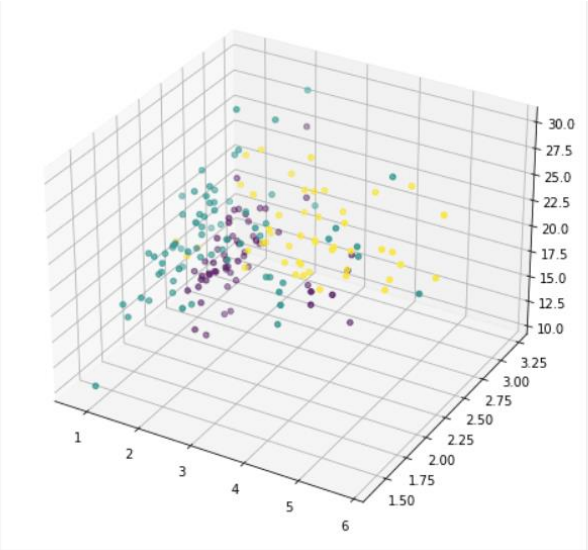

Fig.2. Any three to Among the 13 features and plot with matplotlib (color-coded the different classes using the c argument) i.e; the above diagram in 3-dimension

But this doesn't reveal much of how the data looks like, because the majority of the features are not shown. So We now resort to principal component analysis:

Model Creation

Here we transform the input data X by PCA into Xt. We consider only the first two columns, which contain the most information, and plot it in two dimensional. We can see that the purple class is quite distinctive, but there is still some

overlap. If we scale the data before PCA, the result would be different:

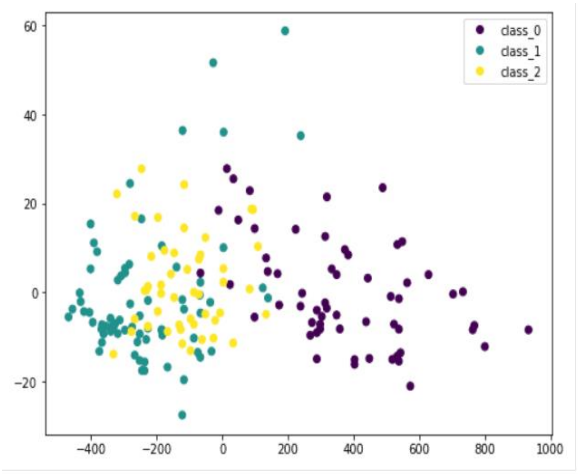

Fig.3. applying PCA on the 3 features used above. showing the transformation the input data X by PCA into Xt

Here we transform the input data X by PCA into Xt. We consider only the first two columns, which contain the most information, and plot it in two dimensional. We can see that the purple class is quite distinctive, but there is still some overlap. If we scale the data before PCA, the result would be different

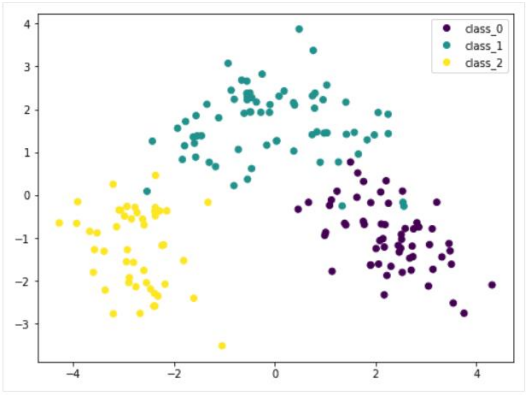

Fig.4.result after we normalized each feature by StandardScaler showing the different classes are more distinctive

Because PCA is sensitive to the scale, if we normalized each feature by StandardScaler we can see a better result. Here the different classes are more distinctive. By looking at this plot, we are confident that a simple model such as SVM can classify this dataset in high accuracy.

Putting these together, the following is the complete code to generate the visualizations.

If we apply the same method on a different dataset, such as MINST handwritten digits, the scatterplot is not showing distinctive boundaries and therefore it needs a more complicated model such as a neural network to classify:

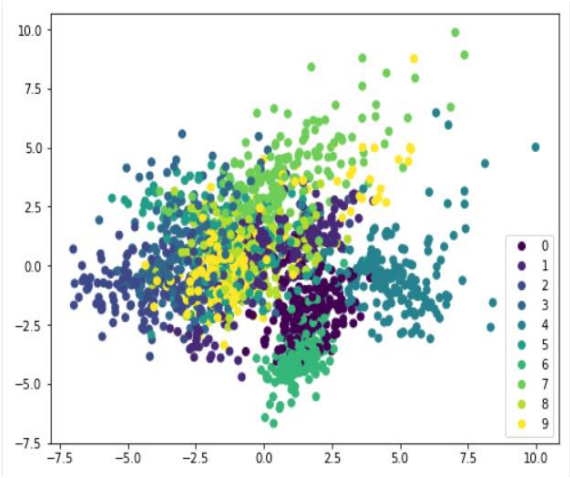

Fig.5 Showing the result after applying the above model on different different dataset, such as MINST handwritten digits,

.

II. Visualizing the explained variance

PCA in essence is to rearrange the features by their linear combinations. Hence it is called a feature extraction technique. One characteristic of PCA is that the first principal component holds the most information about the dataset. The second principal component is more informative than the third, and so on.

To illustrate this idea, we can remove the principal components from the original dataset in steps and see how the dataset looks like. Let’s consider a dataset with fewer features, and show two features in a plot

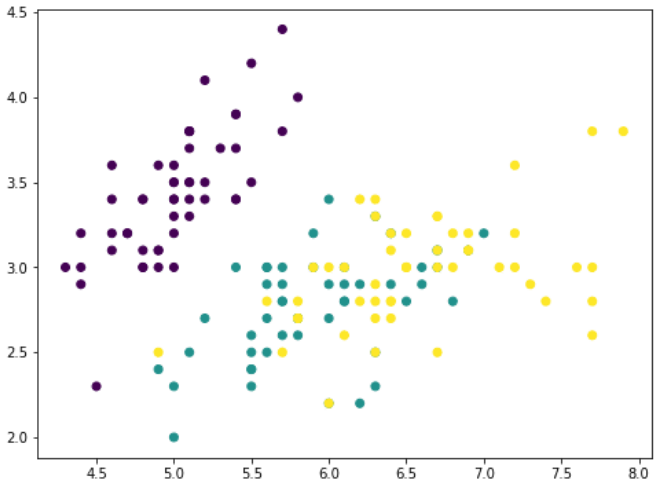

Fig. 6 illustrates this idea principal component holds the most information about the dataset. work is done by using irish dataset.

This is the iris dataset which has only four features. The features are in comparable scales and hence we can skip the scaler. With a 4-features data, the PCA can produce at most 4 principal components they are as follows.

```
[[ 0.36138659 -0.08452251  0.85667061  0.3582892 ]
 [ 0.65658877  0.73016143 -0.17337266 -0.07548102]
 [-0.58202985  0.59791083  0.07623608  0.54583143]
 [-0.31548719  0.3197231   0.47983899 -0.75365743]]
```

For example, the first row is the first principal axis on which the first principal component is created. For any data point p with features

p = (a, b, c, d), since the principal axis is denoted by the vector(v) where v,

v=(0.36 * a-0.08 * b + 0.86, 0.36), the first principal component of this data point has the value 0.36*a-0.08*b+0.86*c+0.36*d on the principal axis. Using vector dot product, this value can be denoted by -> (p.v)

Therefore, with the dataset X has a 150 * 4 matrix (150 data points, each has 4 features), we can map each data point into to the value on this principal axis by matrix-vector multiplication:

i.e; X . v

and the result is a vector of length 150. Now if we remove from each data point the corresponding value along the principal axis vector, that would be :

X – (X . V) . V(transpose)

where the transposed vector v(transpose) is a row and X . v is a column. The product (X . v) v(transpose) follows matrix-matrix multiplication and the result is a 150 * 4 matrix, same dimension as X.

If we plot the first two feature of (X . v) . v(tranpose), it looks like this:

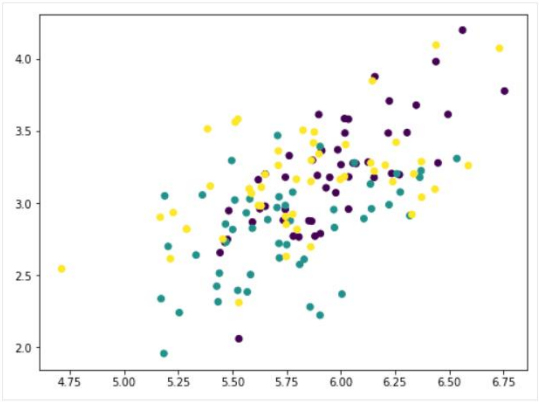

Fig. 7 showing the scattering pattern of to two feature of (X . v) . v(transpose).

The numpy array Xmean is to shift the features of X to centered at zero. This is required for PCA. Then the array value is computed by matrix-vector multiplication.

The array value is the magnitude of each data point mapped on the principal axis. So if we multiply this value to the principal axis vector we get back an array pc1. Removing this from the original dataset X, we get a new array Xremove. In the plot we observed that the points on the scatter plot crumbled together and the cluster of each class is less

distinctive than before. This means we removed a lot of information by removing the first principal component. If we repeat the same process again, the points are further crumbled.

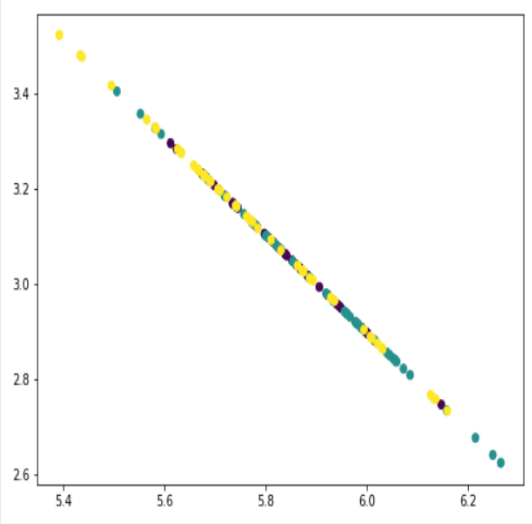

Fig. 8 showing we removed a lot of information by removing the first principal component. If we repeat the same process again, the points are further crumbled:

This looks like a straight line but actually not. If we repeat once more, all points collapse into a straight line.

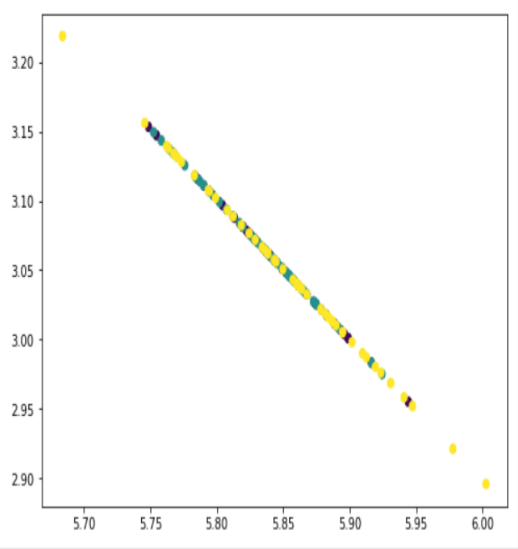

Fig.10 showing the above conclusion that . If we repeat the same process again, the points are further crumbled:

The points all fall on a straight line because we removed three principal components from the data where there are only four features. Hence our data matrix becomes rank 1. You can try repeat once more this process and the result would be all points collapse into a single point. The amount of information removed in each step as we removed the principal components can be found by the corresponding explained variance ratio from the PCA:

print(pca.explained_variance_ratio_)

[0.92461872 0.05306648 0.01710261 0.00521218]

Here we can see, the first component explained 92.5% variance and the second component explained 5.3% variance. If we removed the first two principal components, the remaining variance is only 2.2%, hence visually the plot after removing two components looks like a straight line. In fact, when we check with the plots above, not only we see the points are crumbled, but the range in the x- and y-axes are also smaller as we removed the components.

In terms of machine learning, we can consider using only one single feature for classification in this dataset, namely the first principal component. We should expect to achieve no less than 90% of the original accuracy as using the full set of features:

Using all features, accuracy: 1.0

Using all features, F1: 1.0

Using PC1, accuracy: 0.94

Using PC1, F1: 0.9332591768631814

The other use of the explained variance is on compression. Given the explained variance of the first principal component is large, if we need to store the dataset, we can store only the the projected values on the first principal axis (X .v), as well as the vector 'v' of the principal axis. Then we can approximately reproduce the original dataset by multiplying them:

X is euivalaent to (X.v).v(transpose)

In this way, we need storage for only one value per data point instead of four values for four features. The approximation is more accurate if we store the projected values on multiple principal axes and add up multiple principal components.

Putting these together, the following is the complete code to generate the visualizations:.

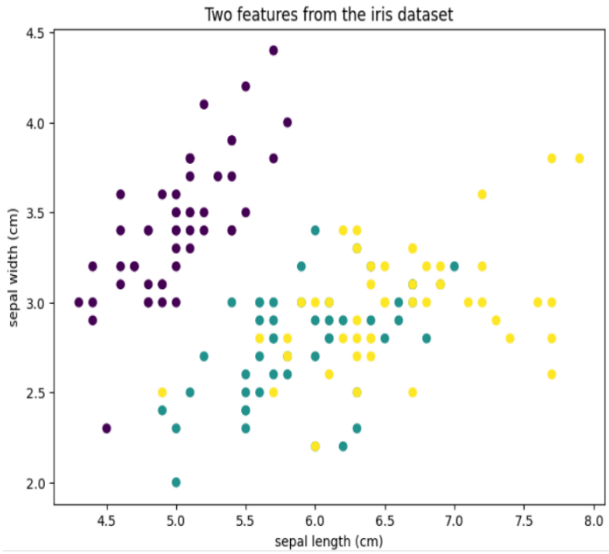

Fig 11. Showing then features the scatter pattern of two features taken from iris dataset

```
Show the principal components
```

Principal components:

[[0.36138659 -0.08452251 0.85667061 0.3582892]
 [0.65658877 0.73016143 -0.17337266 -0.07548102]
 [-0.58202985 0.59791083 0.07623608 0.54583143]
 [-0.31548719 0.3197231   0.47983899 -0.75365743]]

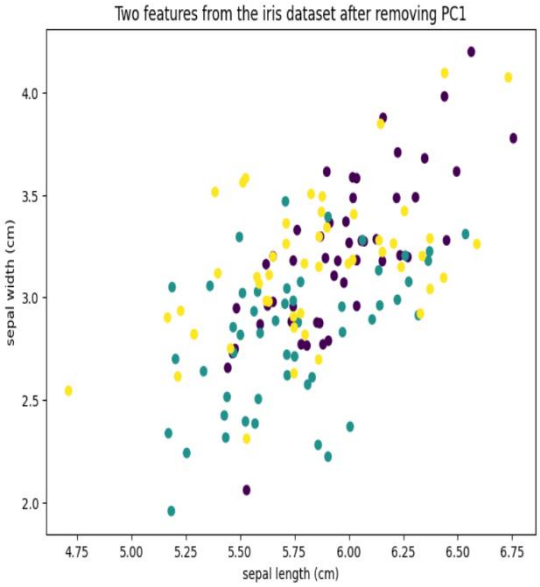

Fig 12. Showing the scattering plot after removing the PC1(Principle component 1)

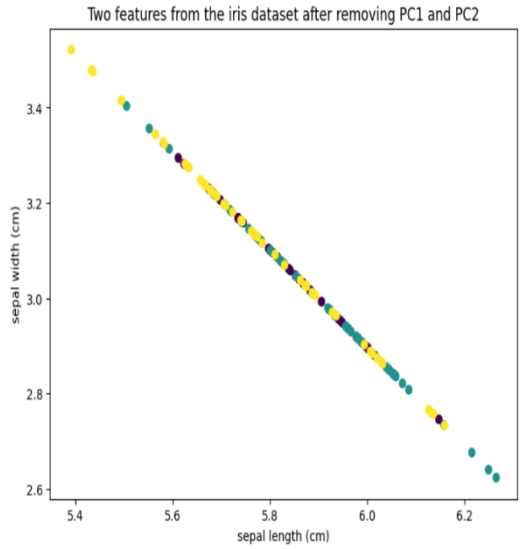

Fig 13. showing the scattering plot after removing PC1 and PC2

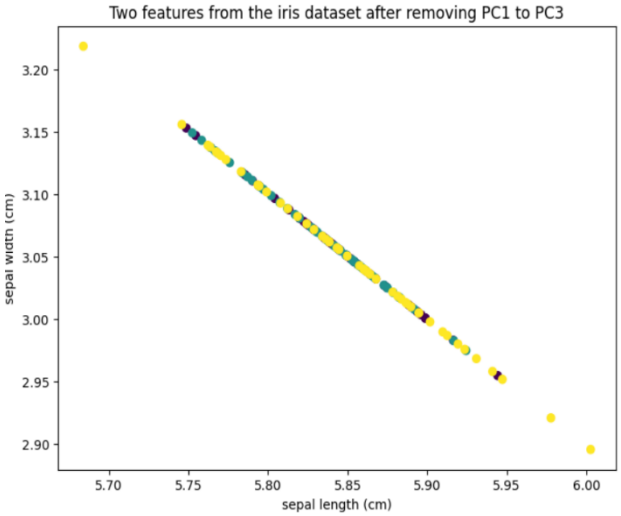

Fig 14. Showing the Scattering plot after removing PC1 and PC3

Explained variance ratios:
[0.92461872 0.05306648 0.01710261 0.00521218]
Using all features, accuracy: 0.98
Using all features, F1: 0.980238302818948
Using PC1, Accuracy: 0.94
Using PC1, F1: 0.9411255411255411

The datasets used in the paper include:

MINST Handwritten Digits:

A dataset featuring handwritten digits.

Used to illustrate the limitations of traditional 2D and 3D visualizations and the need for more complex models like neural networks.

the iris dataset (classification).

The iris dataset is a classic and very easy multi-class classification dataset.

Classes 3

Samples per class 50

Samples total 150

Dimensionality 4

Features real, positive

Discussion

1. Utilization and Significance of PCA:

PCA serves as a critical tool in the realm of unsupervised learning, particularly renowned for its pivotal role in dimensionality reduction and data visualization. Its significance lies in its ability to reveal the underlying structure of high-dimensional data by identifying the principal components, which capture the most variance.

2. Visualizing High-Dimensional Data:

PCA's application in visualizing high-dimensional data provides a transformative insight into complex datasets. The limitations of traditional 2D and 3D visualizations are evident, and PCA's ability to condense data into fewer dimensions allows for clearer and more comprehensible visual representations.

3. Explained Variance and Dataset Structure:

The concept of explained variance within PCA plays a crucial role in understanding the dataset's structure. The step-by-step removal of principal components showcases the proportional loss of information and its correlation with the explained variance ratios. This highlights the impact of removing components on the dataset's integrity and dimensionality.

4. Impact on Machine Learning Models:

The practical application of PCA in machine learning tasks is notable. By reducing dimensionality while retaining crucial information, PCA aids in enhancing model efficiency and accuracy. The demonstrated capability of PCA to facilitate classification with fewer features underscores its value in optimizing machine learning models.

5. Practical Implications and Limitations:

While PCA offers remarkable benefits in data analysis and preprocessing, it's sensitive to scaling, and its efficacy varies depending on the dataset. Understanding the limitations of PCA is crucial for its appropriate application in different contexts.

6. Real-World Relevance:

Demonstrating PCA's effectiveness with real-world datasets, such as the wine dataset and the MINST handwritten digits, solidifies its practical relevance. The visible impact on class boundaries, information retention, and subsequent model performance reinforces its utility in diverse data analysis scenarios.

II. Conclusion

The paper demonstrates the pivotal role of Principal Component Analysis (PCA) in transforming high-dimensional data into comprehensible visualizations. It elucidates the challenges of visualizing complex datasets, emphasizing PCA's ability to reduce dimensions for enhanced data representation. Using real-world datasets like the wine dataset, the study showcases PCA's transformative power in rendering distinct class boundaries, enabling clear classification. The sensitivity of PCA to data scaling is highlighted, influencing the quality of visualizations.

In the wine dataset, employing PCA reduced 13 dimensions into 2 principal components, effectively enabling a clearer understanding of class separation. By depicting this in scatter plots, the results show distinct class boundaries, allowing for efficient classification. The graphs display the impact of PCA on transforming the dataset into lower dimensions, showcasing the reduction in dimensionality and improved visualization compared to conventional 2D and 3D plotting.

Additionally, the exploration into explained variance within PCA reveals the stepwise removal of principal components and its impact on the dataset's structure. The graphs show the proportional loss of information with the removal of principal components, correlating with the explained variance ratios. As principal components are eliminated, the scatter plot points converge, indicating a reduction in dataset information. This highlights the necessity of retaining principal components to preserve the dataset's structure and integrity.

The study also highlights the practical implementation of PCA in machine learning, showcasing its ability to maintain

substantial dataset information, enabling efficient classification with fewer features. The accuracy of classification models with PCA-processed data is notably high, emphasizing the efficacy of PCA in preserving information vital for classification tasks.

Overall, the paper's graphical representations visually affirm PCA's efficacy in reducing dimensions for improved visualization and its profound impact on dataset structure and machine learning model accuracy.

ACKNOWLEDGEMENTS

We would like to express our heartfelt gratitude to Dr. Usha Mittal for her invaluable contributions and unwavering support throughout this project. Her expertise, guidance, and dedication have been instrumental in its success. We are deeply thankful for her mentorship and collaboration.

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, pp. 436-444, 28 May 2015.

[2] "Large Scale Visual Recognition Challenge (ILSVRC) – ImageNet," ImageNet, http:// www.image-net.org/challenges/LSVRC.

[3] D.G. Lowe, "Distinctive image features from scale-invariant keypoints," Int. Journal of Computer Vision, vol. 60, no. 2, pp. 91-110, 2004.

[4] D. Navneet and B. Triggs, "Histograms of oriented gradients for human detection," IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 886–893, 2005.

[5] Y. LeCun et al., "Backpropagation applied to handwritten zip code recognition," Neural Computation, vol. 1, no. 4, pp. 541-551, 1989.

[6] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," NIPS: Neural Info. Proc. Sys., Lake Tahoe, Nevada, 2012.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv 1409.1556, Sep 2014.

[8] C. Szegedy et al., "Going deeper with convolutions," arXiv 1409.4842, Sep 2014.

[9] C. Szegedy et al., "Rethinking the Inception Architecture for Computer Vision," IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[10] K. He et al., "Deep residual learning for image recognition," arXiv 1512.03385, Dec 2015.

[11] G. Huang, "Dense connected convolutional neural networks," IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[12] "TensorFlow: An open-source software library for machin intelligence," TensorFlow, https://www.tensorflow.org/.

[13] F. Chollet at al., "Keras", GitHub, 2017, https://github.com/fchollet/keras.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 9, pp. 19041916, Sep 2015.

[15] "UC Merced Land Use Dataset," University of California, Merced, http://weegee.vision.ucmerced.edu/datasets/landuse.html.

[16] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," Proc. 18th ACM SIGSPATIAL International Symposium on Advances in Geo. Info. Sys., pp. 270-279, 3-5 Nov 2010.

[17] Y. Liang, S. Monteiro, and E. Saber, "Transfer learning for highresolution aerial image classification," IEEE Workshop Applied Imagery Pattern Recognition (AIPR), Oct 2016.

[18] M. Castelluccio, G. Poggi, and L. Verdoliva, "Land Use Classification in Remote Sensing Images by Convolutional Neural Networks," arXiv 1508.00092, Aug 2015.

[19] G. Scott, M. England, W. Starms, R. Marcum, and C. Davis, "Training deep convolutional neural networks for land–cover classification of high-resolution imagery", IEEE Geoscience and Remote Sensing Letters, vol. 14, no. 4, pp. 549-553, Apr 2017.

[20] "SpaceNet on AWS," Amazon.com, https://aws.amazon.com/publicdatasets/spacenet/.

[21] E. Chartock, W. LaRow, and V. Singh, "Extraction of Building Footprints from Satellite Imagery," Stanford University Report, 2017.

[22] G. Cheng, J. Han, and X. Lu, "Remote Sensing Image Scene Classification: Benchmark and State of the Art," Proc. IEEE, vol. 105, no. 10, pp. 1865-1883, October 2017.

[23] "Functional Map of the World Challenge," IARPA, https://www.iarpa.gov/challenges/fmow.html.

[24] "Marathon Match: Functional Map of the World," TopCoder, https://community.topcoder.com/longcontest/?module=ViewProblemStatement&rd=16996&compid=57158.

[25] "Earth on AWS: Functional Map of the World," Amazon.com, https://aws.amazon.com/earth/.

[26] G. Christie. N. Fendley, J. Wilson, and R. Mukherjee, "Functional map of the world," arXiv 1711.07846, 21 Nov 2017.

[27] F. Chollet, "Xception: deep learning with depthwise separable convolutions," arXiv 1610.02357, Oct 2016.

[28] C. Ju, A. Bibaut, and M. van der Laan, "The relative performance of ensemble methods with deep convolutional neural networks for image classification," arXiv 1704.01664, April 2017.

[29] "Applications", Keras, https://keras.io/applications/.

[30] F. Yu, "CNN Finetune", Github, 2017, https://github.com/flyyufelix/cnn_finetune.

[31] A. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition," IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 512-519, 2014.

[32] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" Proc. 31st Int. Conf. Machine Learning, vol. 32, pp. 647-655, 2014.