PCA and Autoencoders

INSE 6220 – ADVANCED STATISTICAL APPROACHES TO QUALITY
Srikanth Amudala
Concordia University

***Abstract* –This paper presents the implementation and Analysis of two different techniques of dimensionality reduction for a given dataset. One being linear(PCA) and other being non-linear(Autoencoder). The analysis involves the implementation of an autoencoder and PCA for reconstructing the dataset using the MINST [4] dataset and the Wine dataset [3].**

***Keywords*—Principal Components Analysis (PCA), Autoencoder, Image processing, data MNIST.**

I. INTRODUCTION

In terms of performance, having data of high dimensionality is problematic because (a) it can mean high computational cost to perform learning and inference and (b) it often leads to overfitting [1] when learning a model, which means that the model will perform well on the training data but poorly on test data. Dimensionality reduction addresses both of these problems, while (hopefully) preserving most of the relevant information in the data needed to learn accurate, predictive models. It is a technique or a process for reducing the number of dimensions in a dataset into a compressed representation called its principal components or variables under consideration used in statistics, machine learning, and information theory. There are various ways to explore the state-of-the-art on dimensionality reduction techniques transforming the data in the high-dimensional space to a space of fewer dimensions such as Principal Component Analysis, Non-negative matrix factorization, Kernel PCA, Graph-based kernel PCA, Linear discriminant analysis (LDA), Generalized discriminant analysis (GDA) and autoencoders. This project presents an exploration of the steps and analysis of the intermediate results of PCA and Autoencoders.

Principal Component Analysis (PCA) is a very popular technique for dimensionality reduction. Given a set of data on 'n' dimensions, PCA aims to find a linear subspace of dimension 'd' lower than 'n' such that the data points lie mainly on this linear subspace. Such a reduced subspace attempts to maintain most of the variability of the data. PCA is used to merge information in a dataset which can be elaborated by multiple variable.

Autoencoder is a type of artificial neural network used to learn efficient data coding's in an unsupervised manner. The aim of an auto-encoder is to learn a representation (encoding) for a set of data, typically for dimensionality reduction. Along with the reduction side, a reconstructing side is learnt, where the autoencoder tries to generate from the reduced encoding a representation as close as possible to its original input, hence its name. [2] Recently, the autoencoder concept has become more widely used for learning generative models of data. Some of the most powerful AI in the 2010's have involved sparse autoencoders stacked inside of deep neural networks.

II. DATASET

In order to analyse these two-dimension reduction techniques we have used two different datasets, one with a feature set of 786(Mnist handwritten digits) and the other with 13 (Wine recognition dataset)features.

A. Wine recognition dataset [3]

The data is the results of a chemical analysis of wines grown in the same region in Italy by three different cultivators.
Data Set Characteristics: Number of Instances: 178 (50 in each of three classes.)

Number of Attributes: 13 Numeric

- *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.
- *Class(Target Values)*: class_0, class_1, class_2.

Table 1: Summary Statistics for the 3 types of wines

Sr.No.	Attribute Information	Min	Max	Mean	SD
1	Alcohol	11	14.8	13	0.8
2	Malic Acid	0.74	5.8	2.34	1.12
3	Ash	1.36	3.23	2.36	0.27
4	Alcalinity of Ash	10.6	30	19.5	3.3
5	Magnesium	70	162	99.7	14.3
6	Total Phenols	0.98	3.88	2.29	0.63
7	Flavanoids	0.34	5.08	2.03	1
8	Non Flavanoids Phenols	0.13	0.66	0.36	0.12
9	Proanthocyanins	0.41	3.58	1.59	0.57
10	Colour Intensity	1.3	13	5.1	2.3
11	Hue	0.48	1.71	0.96	0.23
12	OD280/OD315 of diluted wines	1.27	4	2.61	0.71
13	Proline	278	1680	746	315

Table 1

For better understanding of the data, Fig. 1 Plot's the Box Plot showing the distribution of each feature in the database. From the plot, it can be observed that 50% of the data varies in from the range -1 to 1 and there are seven variables with outliers i.e. Malic acid, ash, alcalinity of ash, magnesium, proanthocyanins, colour intensity, hue. In addition, variables such as, malic acid, magnesium, Total phenols, Flavanoids, Nonflavanoid phenols, Proanthocyanins, Colour intensity, and proline are positively skewed and ash is negatively skewed.

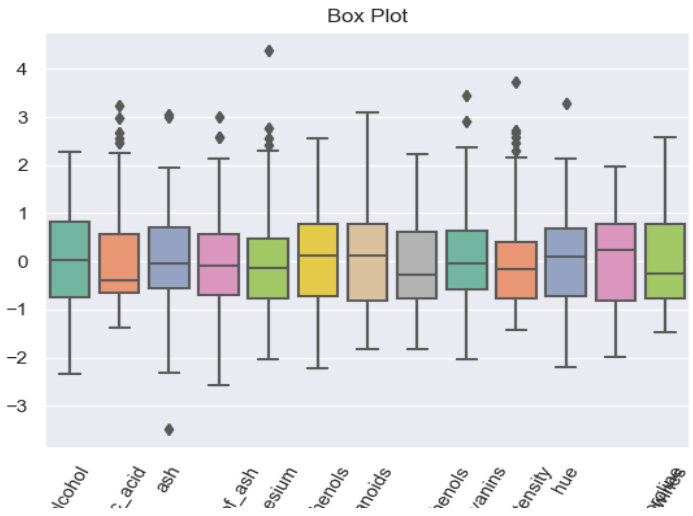


Fig. 1. Box Plot of the features of Wine Dataset

Fig. 2. Shows the pair wise relationship between the 13 features corelating with each other.

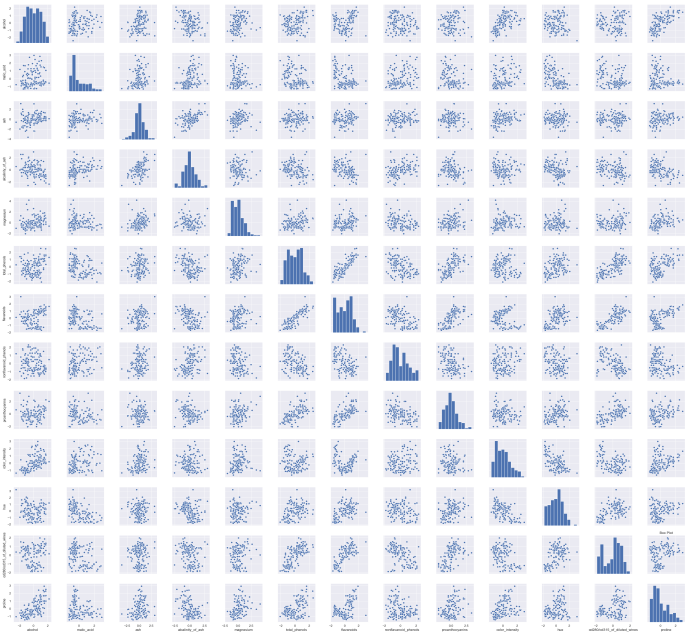


Fig. 2. Pair wise relationships of the Dataset

B. MNIST Dataset[4]

This Dataset consists of 10 digits of 60,000 training images having 28x28 grayscale is chosen to apply auto encoder reduction techniques, along with a test set of 10,000 testing images. It is a subset of a larger set available from NIST. The data has been made by "re-mixing" the samples from NIST's original datasets. Also, the black and white images taken from NIST are normalized so that it can get to fit into a 28x28 pixel box and are anti-aliased to get grayscale levels. The digits have been size-normalized and centered in a fixed-size image. Fig. 3 shows the sample hand written images from the dataset.

Fig. 3. Sample data of hand-written digits

IMAGE CLASSES AND LABELS

CLASSES	LABELS
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9

Table 2

III. MODELLING TECHNIQUES

A. Principal Component Analysis

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables (entities each of which takes on various numerical values) into a set of values of linearly uncorrelated variables called principal components. If there are n observations with p variables, then the number of distinct principal components is min(n-1, p). [5]

In general data matrix X, which is a n × p (178x13) matrix will be pre- processed before the PCA analysis. First, it will be zero centered, which means the mean of each column will be equal to zero. We have zero centered the Wine data and calculated the covariance matrix **S**, which is shown in Figure 5.

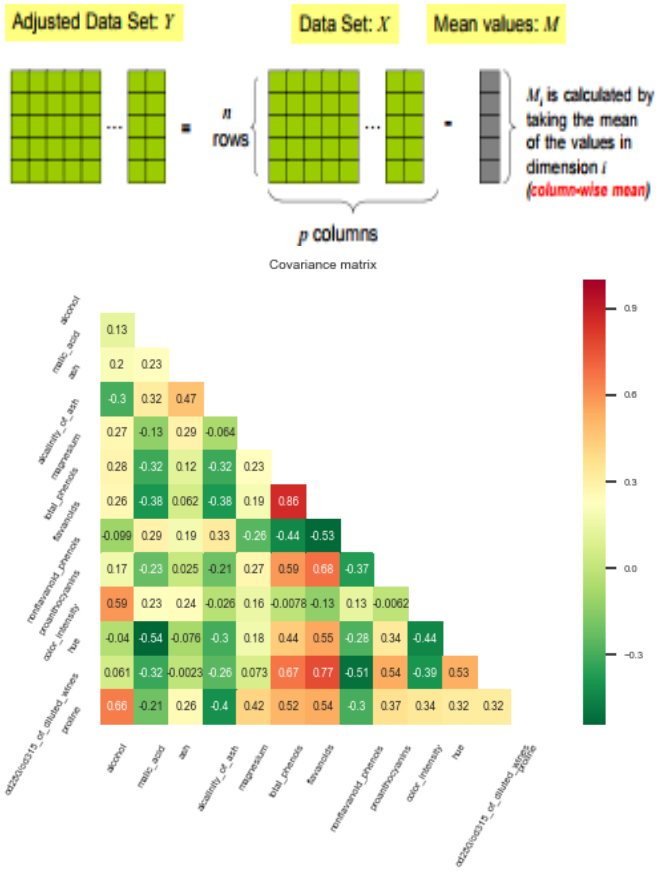


Fig. 5. Covariance Matrix of zero centered data

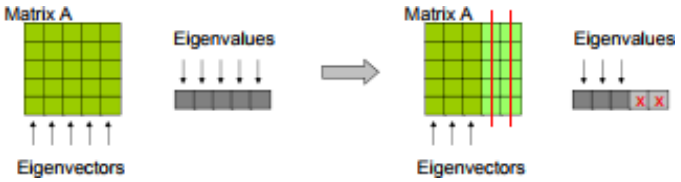
Using the Covariance Matrix from Fig. 5, we can observe the features that are corelated to each other. From the plot we can observe :

- Features total_phenols, Flavanoids are positively corelated to features OD280/OD315 of diluted wines and Proline respectively.
- Flavanoids and total_phenols are highly corelated to each other.

Next step in PCA analysis is to implement the eigen value decomposition of the covariance matrix S, as follows:

$$S = A \Lambda A^{T} \quad (1)$$

where A is a p × p matrix of eigenvectors, and Λ is a diagonal matrix of eigenvalues. We have calculated the eigenvalues and eigenvectors of the covariance matrix for the bank data.



Where A = (a_1 , a_2, . . . , a_p) is a p × p orthogonal matrix (A ′A = I) where columns are denoted by a_j = (a_{j1} , a_{j2} , . . . , a_{jp}) ′ are the eigenvectors of S such that a ′ j a_j = 1, j = 1, . . ., p. The eigenvectors depicts the linear combination of existing data. The first eigenvector a1 defines a variable having the most data variability. The eigenvectors are the principal component (PC) coefficients, also known as loadings. Hence, each eigenvector a_j contains coefficients for the j^{th} PC.

Fig. 6, Fig. 7 shows the Scree plot and Pareto plot of the Explained Variance, explained variance is the amount of variance accounted for by: 1st principal component, λ1, 1st eigenvalue 2nd principal component, λ2, 2nd eigenvalue 4 ...

$$\lambda 1 > \lambda 2 > \lambda 3 > \lambda 4 > \ldots > \lambda p$$

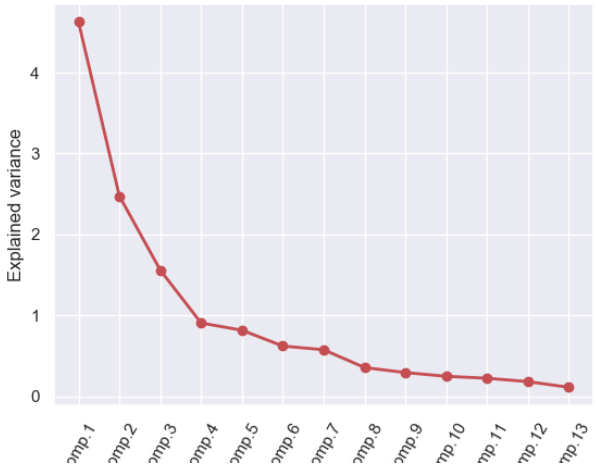


Fig. 6. Scree plot of explained variance

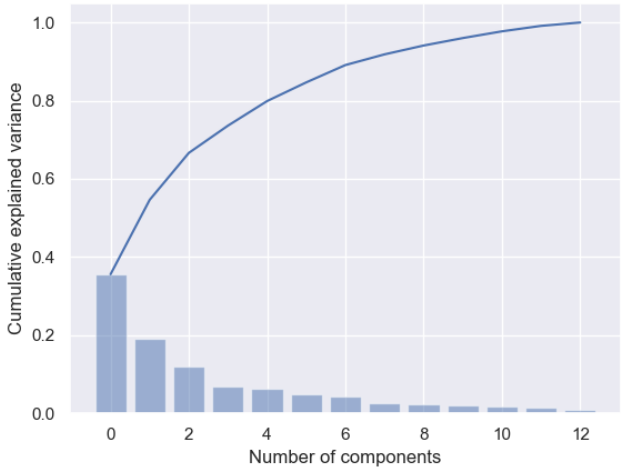


Fig. 7. Pareto plot of explained variance

Based on the explained variance by both the PC1 and PC2 and also from the Scree and Pareto plots, it can be deduced that the lowest-dimensional space to represent the Wine Dataset corresponds to d = 2 or 3. For better analysis and plotting we have considered d = 2.

After calculating the principal components, there can be plotted against each other to have a better insight about their relationships. Fig. 8 shows the plots.

Another important aspect of principal component analysis is that how much each attribute contributes to principal components. For this purpose, PC coefficients are plotted against each other in Figure 9. The following items are some of the facts that can be inferred from these plots:

- Malic_acid and color_intensity are the feature which contributes to PC1 more than others

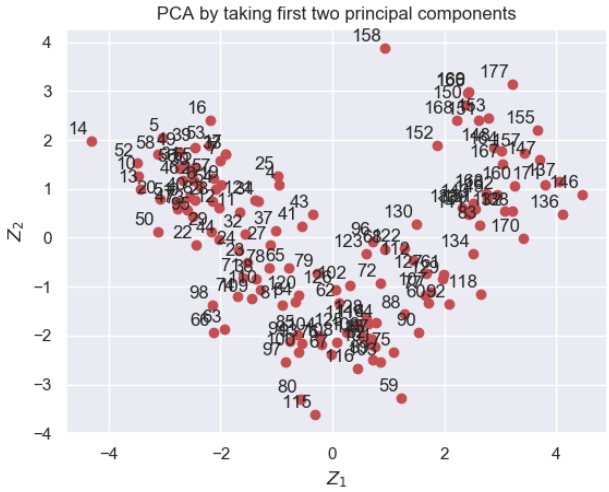


Fig. 8. PCA by taking first two principal components

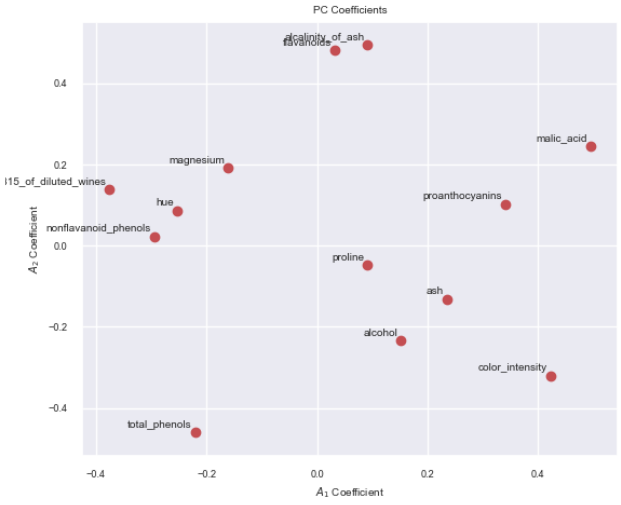


Fig. 9. Scatter plot of PC1 and PC2 coefficients

When the goal is to extract important information from the data matrix, the problem is to figure out how many components should be considered. There are different solutions to this problem. For instance, we can keep components whose eigen values are larger than the average eigen value. Another criteria which is used in this work is to calculate the explained variance for each component as follows:

$$\ell_j = \frac{\lambda_j}{\sum_{j=1}^{p} \lambda_j} \times 100\%, \quad j = 1, ..., p. \tag{2}$$

Using the above equation, we have calculated the explained variance for the components in Wine data set, and the result is as follows

l_1=36.4243%, l_2=18.9289%, l_3=11.2175%, l_4=7.6758%, l_5=6.32%, l_6=4.7496%, l_7=4.1341%, l_8=2.6921%, l_9=2.1857%, l_{10}=2.094%, l_{11}=1.5527%, l_{12}=1.3082%, l_{13}=0.7096%

Based on these values, which are also presented in Scree plot Figure 6, we can infer that the first two components account for 55% of data and the summation of first 3 components account for 66.5% of data and the summation of first 4 components account for 74% of data, the summation of first 5 components account for 80.5% of data. We can now choose the first 4 to 5 component's to have the maximum amount of data.

From the Pareto plot in Figure 7, the cumulative explained variance for the components can be inferred. As it can be easily observed, the first 3 components account for most of the variance in data. Another useful plot in PCA is biplot. This plot shows two kind of information at the same time: first it shows the PCA scores for observations, and second it shows how much each original variable contributes to the principal components. Observations are presented as points in the plot, and variables are presented as vectors. The axis of bi-plot are the principal components. The 2D bi-plots for the Wine data set is presented in Fig. 10. The following information can be inferred from the plot:

- Alcohol, Magnesium, proline, total phenols, lavanoids, hue etc are positively corelated
- Malic_acid, alkalinity of ash, nonflavanoid_phenols etc are negatively corelated.

Fig. 11. Describes the Principal Components of all the features. The following information can be inferred from the plot:

- In PC1, Feature Flavanoids accounts for most of the data.
- In PC2, Features Alcohol and Color intensity accounts for most of the data.

- In PC3, malic acid and Magnesium accounts for most of the data.

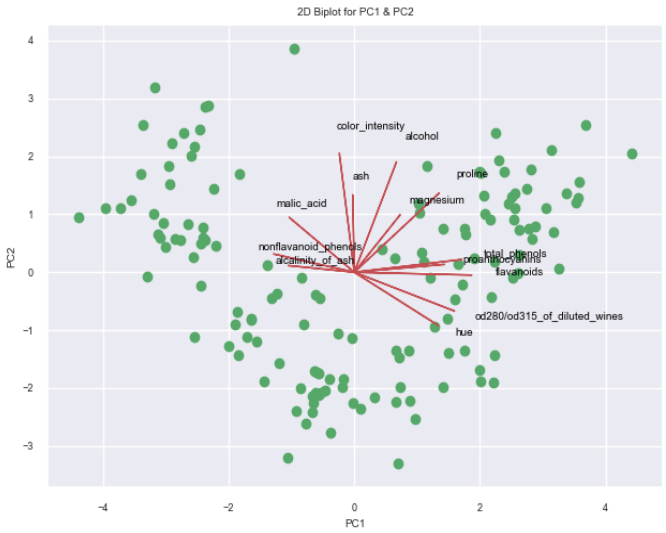


Fig. 10. 2D Biplot for PC1 & PC2

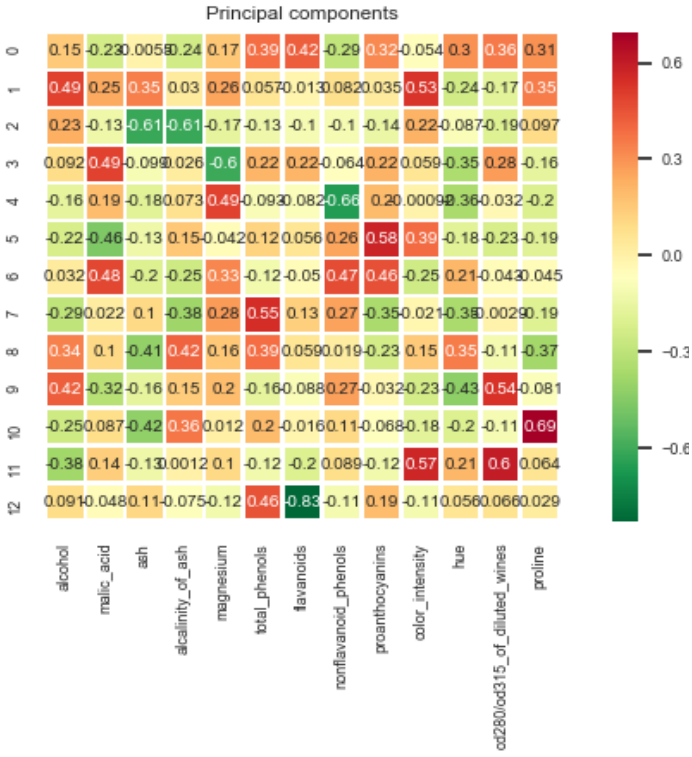


Fig. 11. Principal Components

B. Auto Encoder:

An autoencoder is a type of artificial neural network used to learn efficient data encodings in an unsupervised manner. The aim of an autoencoder is to learn a representation (encoding) for a set of data, typically for dimensionality reduction. Along with the reduction side, a reconstructing side is learnt, where the autoencoder tries to generate from the reduced encoding a representation as close as possible to its original input [6],

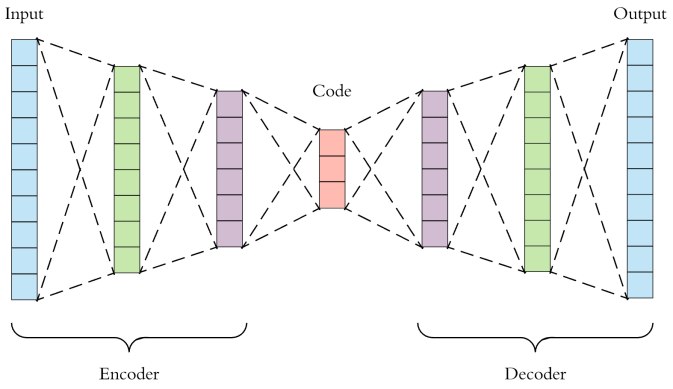


Fig. 12. Autoencoder architecture [7]

Using a simple autoencoder architecture described in Figure 12, the training set, X, can be encoded (compressed) by training X to learn itself. The encoding layer is chosen to have 4 input dimensions with ReLU activation function. The size of this layer was chosen arbitrarily but mainly based on neural networks trained with the classic MNIST dataset in the Keras documentation for autoencoders [8].

This results in a dimension reduction from 784 features to only 4 features. We choose this on purpose to see how well the autoencoder performs by reducing the features by 196% ~ 200%.

Following the training of the autoencoder, we can visualize how it is reconstructing the image data by encoding and decoding images from the test set. A sample of these results can be seen in Figure 13.

- The first row represents the original images with 784 features.
- The last row represents the respective encoded image which had been reduced to 4 features.
- The middle row represents the reconstructed images from the encoded 4 feature image to 784 feature image by the decoder.

This confirms the autoencoder is in fact reproducing a compressed representation of the information given to it. The basic shapes are learned but details on the edges were not clear because of very less features.

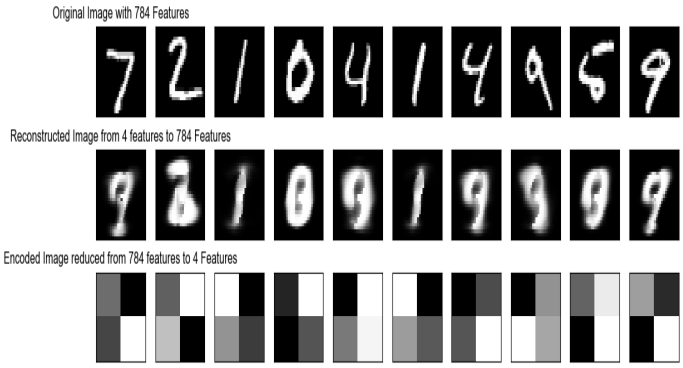


Fig. 13. Autoencoder Intermediate Output

The basic shapes of the data are clearly learned but details like the words on the clothing are not clear.

The decoding layer is then removed from the network and the SoftMax layer is added for training with the training set labels. Once completed, the model performance is evaluated using the test set. Fig. 14 shows the model loss during the training.

Accuracy:

With a dimension reduction from 784 features to 4 features we were able to get an accuracy of 88.82%. Table 3 shows the classification report for the parameter's precision, recall, F1, and support scores for the model.
From Table 3, we can say the precision, recall and F1 are 89% with a support of 10000 test images.

- Precision: We can say that there are 89% of relevant instances among the retrieved instances.
- Recall: 89% of relevant instances have been retrieved over the total amount of relevant instances.

Confusion Matrix:

Fig. 15 shows the confusion matrix for the autoencoder. It is matrix that tells about prediction results on a classification problem by summing up the number of correct and incorrect predictions. As we can see from the Fig. 15 we can clearly see the consistency of the model looking at the true positives being significant for all the classes.

For example for Class 0, the model has predicted 933 True positives and for class 1, the model has successfully predicted 1102 True positives and so on.

The model confused Class 5 with Class 8, Class 3 and predicted 87, 46 False Positives respectively. This could be minimized by increasing the number of dimensions and retaining most of the information.

Class	Precision	Recall	F1-Score	Support
0	0.92	0.95	0.94	980
1	0.95	0.97	0.96	1135
2	0.94	0.84	0.88	1032
3	0.8	0.88	0.84	1010
4	0.89	0.87	0.88	982
5	0.77	0.77	0.77	892
6	0.88	0.95	0.91	958
7	0.89	0.91	0.9	1028
8	0.88	0.79	0.83	974
9	0.88	0.87	0.87	1009
micro avg.	0.89	0.89	0.89	10000
macro avg.	0.89	0.89	0.89	10000
weighted avg.	0.89	0.89	0.89	10000

Table 3

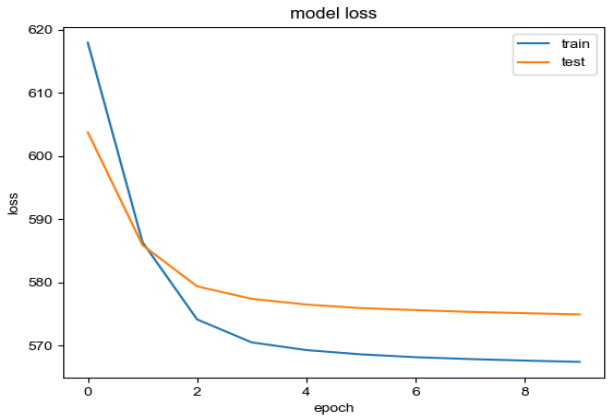


Fig. 14. Model Loss

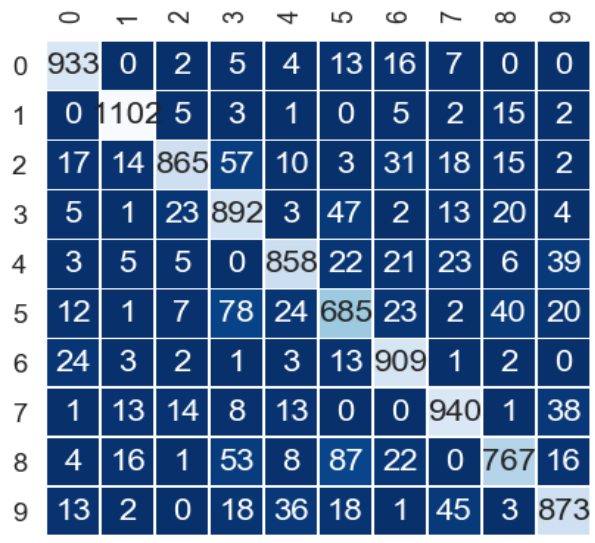


Fig. 15. Confusion matrix of the ratio of the true labels to the predicted labels for each class.

IV. PCA VS. AUTO ENCODER

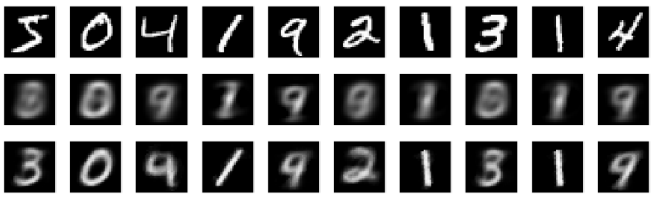


Fig. 16. Original, PCA & Autoencoder image

Fig. 16 shows the reconstructed images from PCA & Autoencoder taking MNIST dataset as input and reducing the dimensions from 784 to 2 as follows:

784→512→128→2→128→512→784

- Row 1 displays the original images of the MNIST dataset.
- Row 2 displays the reconstructed images from PCA respectively.
- Row 3 displays the reconstructed images from Autoencoder respectively.

This shows the inconsistency of PCA, when compared with Autoencoder for the MNIST dataset.

Fig. 17 displays the results when we plot PCA projection side by side with the bottle neck representation by plotting 5000 images clustered with 10 Classes.

PCA reconstruction error with 2 PCs: 0.056

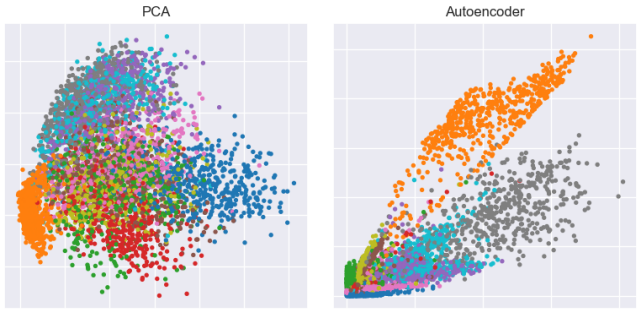


Fig. 17. Bottleneck representation

V. DISCUSSION

This paper investigated about the data analysis with intermediate results and the implementation of the two dimensionality reduction techniques PCA and Auto-encoders with MNIST Dataset and Wine Dataset.

We have successfully reconstructed the images from 2 principal components representing 784 features with a reconstruction error of 0.056 using PCA. We were able to achieve an accuracy of 88.82% using Autoencoders after successfully reducing the input features by nearly 200%.(784 features to 4 features). This allowed us in understanding the concept of dimensionality reduction and its effect on improving the efficiency during the training as well as predicting the labels.

VI. CONCLUSION

PCA is restricted to a linear map, while auto encoders can have nonlinear encoder/decoders. If no non-linear function is used in the Autoencoder and the number of neurons in the hidden layer is of smaller dimension then that of the input then PCA and Autoencoder can yield the same result. Otherwise the Autoencoder may find a different subspace.

REFERENCES

[1]https://moodle.concordia.ca/moodle/pluginfile.php/3385228/mod_resource/content/1/SampleProject.pdf

[2]https://moodle.concordia.ca/moodle/pluginfile.php/3333342/mod_resource/content/1/SampleProject.pdf

[3]https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_wine.html

[4]https://keras.io/datasets/

[5]https://en.wikipedia.org/wiki/Principal_component_analysis

[6] https://en.wikipedia.org/wiki/Autoencoder

[7] https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798