**Available Online at www.ijcsmc.com**

**International Journal of Computer Science and Mobile Computing**

**A Monthly Journal of Computer Science and Information Technology**

**ISSN 2320–088X**
**IMPACT FACTOR: 6.199**

***IJCSMC, Vol. 8, Issue. 12, December 2019, pg.34 – 45***

# FEATURE REDUCTION AND PREDICTION FOR WINE CHEMICAL COMPONENT USING PRINCIPAL COMPONENT ANALYSIS (PCA) AND LINEAR DISCRIMINANT ANALYSIS (LDA)

**[1]Kpigigbue N-Aabe; [2]Orlunwo O. Placida; [3]Friday E. Onuodu**
[1]Computer Science Department, Ignatius Ajuru University of Education, Port Harcourt
[2]Computer Science Department, Ignatius Ajuru University of Education, Port Harcourt
[3]Computer Science Department, University of Port Harcourt
[1] n-aabe.kpigigbue@ust.edu.ng, [2] Orlunwoplacida@gmail.com, [3] gonuodu@gmail.com

***Abstract: High dimensional data is frequently found in various field of study basically in the process of running data analysis; individuals have applied the various techniques available to manage high dimensional data. However, Principal Components Analysis (PCA) and Linear Discriminant Analysis (LDA) have been applied in high dimensional data, reducing the process of classifying features under consideration.***
***Keywords: Dimensional Reduction, Features, Curse, StandardScarler, iloc, and Confusion matrix.***

## 1.1 Introduction

Technological innovations have brought massive data known as Big Data. This has encouraged advancement of computational tools regarding the issue of massive dataset, evokes three principal challenges volume, velocity, variety which are related to dimensions. The curse of dimensionality are all the challenges that an individual encounter in cause of analyzing a high dimensional dataset.

High number of features usually cause sluggish induction process while sending similar outputs as gotten from much smaller feature subset. The purpose of applying dimensionality reduction in machine learning are to improve prediction performance and algorithm efficiency to reduce complexity of learned result. The aim of this paper is to carefully examine prediction accuracy of PCA and LDA as dimensional reduction technique in analyzing a sample data.

 

## 1.2 Principal Component Analysis (PCA)

The reduction of high dimensional data comprised of large number of interrelated variables, while retaining as much as possible of the variation present in the data set is the core brain behind the application of principal component analysis (PCA) I.T. Jolliffe. The idea behind PCA is the challenge of identifying patterns in dataset. Principal component takes charge of highlighting similarities and differences in a high dimensional data to meet graphical representation, and is hinged behind standard deviation, eigenvalues and eigenvectors making PCA a powerful technique used in data analysis Lindsay I Smith.

## 1.3 Linear Discriminant Analysis (LDA)

Linear discriminant produce an uncorrelated number of dataset which can be used maximized for class separation. The discriminant functions effectively more one column of dataset onto the other and for this to be done the target column has to be selected

## 1.4 Related Works

Review on “Implementation of Principal Component Analysis with Fuzzy Annotation for CAD Jewellery Images”, Pinderjeet Kaur applied Principal Component Analysis (PCA) to minimize computational cost of Content Based Image Retrieval (CBIR). The review on Linear and Non-Linear Dimensionality Reduction Techniques study conducted by Arunasakthi. K, KamatchiPriya. L. stated that the profound fundamental and powerful tools for dimensional reduction for retrieving effective features of high dimensional points in input data is the Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA). Julie M. David and Kannan Balakrishanan investigation on dimensional reduction technique principal components generates a set of new variables. The process involves normalization of inputs, computation of the orthonormal vector by PCA, and then the principal components are stacked in descending order of their weight.

Kresimir Delac, Mislav Grgic and Sonja Grgic in their paper “Independent Comparative Study of PCA, ICA, and LDA on the FERET Data Set” found PCA as a dimensional locates and retains the most suitable vector such that the projection of sample does maintain information of original sample. the review “CBIR Feature Vector Dimension Reduction with Eigenvectors of Covariance Matrix using Row, Column and Diagonal Mean Sequences”, Dr. H.B. Kekre, Sudeep D. Thepade and Akshay Maloo noted PCA as a transformational that converts each image to its corresponding eigen-image in the database.

## 1.5 Methodology

The WINE DATASET extracted from UCI Machine Learning data storage showing chemical components grown in Italy but derived from three different cultivars (Ash, Alcohol, Malic acidic) was chosen as a sample dataset due to its simplicity and also deep informative research.

The Wine dataset consists of 172 rows and 13 columns as well as one extra label column which is essentially a group.

Before we start off, let's access our dataset using the Ananconda python development framework

jupyter DimentionalReduction Last Checkpoint: Last Saturday at 11:50 PM (autosaved)

```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt

In [2]: wine = pd.read_csv ("wine.data", header = None)

In [3]: wine.columns = ["Class label", "Alcohol", "Malic acid", "Ash", "Alcalanity of ash",
                        "Magnesium", "Total phenols", "Flavanoids", "Nonflavanoid phenols",
                        "Proanthrocyanins", "Color intensity", "Hue", "OD28/OD315 of diluted wines", "Proline"]

In [6]: wine.head(1000)
```

Out[6]:

| | Class label | Alcohol | Malic acid | Ash | Alcalanity of ash | Magnesium | Total phenols | Flavanoids | Nonflavanoid phenols | Proanthrocyanins | Color intensity | Hue | OD28/OD315 of diluted wines | Proline |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 14.23 | 1.71 | 2.43 | 15.6 | 127 | 2.80 | 3.06 | 0.28 | 2.29 | 5.640000 | 1.04 | 3.92 | 1065 |
| 1 | 1 | 13.20 | 1.78 | 2.14 | 11.2 | 100 | 2.65 | 2.76 | 0.26 | 1.28 | 4.380000 | 1.05 | 3.40 | 1050 |
| 2 | 1 | 13.16 | 2.36 | 2.67 | 18.6 | 101 | 2.80 | 3.24 | 0.30 | 2.81 | 5.680000 | 1.03 | 3.17 | 1185 |
| 3 | 1 | 14.37 | 1.95 | 2.50 | 16.8 | 113 | 3.85 | 3.49 | 0.24 | 2.18 | 7.800000 | 0.86 | 3.45 | 1480 |
| 4 | 1 | 13.24 | 2.59 | 2.87 | 21.0 | 118 | 2.80 | 2.69 | 0.39 | 1.82 | 4.320000 | 1.04 | 2.93 | 735 |
| 5 | 1 | 14.20 | 1.76 | 2.45 | 15.2 | 112 | 3.27 | 3.39 | 0.34 | 1.97 | 6.750000 | 1.05 | 2.85 | 1450 |
| 6 | 1 | 14.39 | 1.87 | 2.45 | 14.6 | 96 | 2.50 | 2.52 | 0.30 | 1.98 | 5.250000 | 1.02 | 3.58 | 1290 |
| 7 | 1 | 14.06 | 2.15 | 2.61 | 17.6 | 121 | 2.60 | 2.51 | 0.31 | 1.25 | 5.050000 | 1.06 | 3.58 | 1295 |

(IDE) "JUPYTER" to load our data

*Figure 1:Loading of dataset (source: Juypter notebook)*

Figure 1 is the sample dataset representation providing the different chemical components of three classifiers (group labels) in terms of alcohol, malic and Ash.

## 1.6 Training, Testing and Visualization of PCA and LDA

The column of interest is the first column which has the 'Group label'. iloc is used for index's position. The rest columns include the 13 ingredients found in wine.

```
In [11]: x = wine.iloc [:, 1:].values
         y = wine.iloc[:, 0].values
```

*Figure 2: using iloc for index position*

After defining our X and Y, splitting takes place for training and testing.

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split (x, y, test_size= 0.2, random_state= 0)
```

*Figure 3: Splitting dataset*

Dataset is normalized (or standardize) using StandardScaler from sklearn.

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
x_train = scaler.fit_transform(x_train)
x_test = scaler.transform(x_test)
```

*Figure 4: Standardizing dataset using StandardScaler*

Dataset is ready for PCA LDA process after standardizing all the features.

## 1.7 Testing and Training for PCA

```
#Importing PCA from scikit learn

from sklearn.decomposition import PCA
```

*Figure 5: Using PCA from sklearn*

Using 2 components for number of PCAs.

```
#We can now pass in the number of principal components; let's choose 2 components.

pca = PCA (n_components = 2)
#training

x_train = pca.fit_transform(x_train)
#testing
x_test = pca.transform(x_test)
```

*Figure 6: Passing PCA components*

Figure 7 showing the explained variance ratio.

```
In [12]: #We can take a look at the explained variance ratio.

         explained_variance = pca.explained_variance_ratio_
         explained_variance

         #The first component accounts for 36.9% of
         #the variance while the second component accounts for 19.3%.

Out[12]: array([0.36884109, 0.19318394])
```

*Figure 7: Variance ratio*

36.9% accounts for first component variance and 19.3% accounts for the second.

Running logistic regression to the training set.

 

```
In [13]: #Now, let's fit a logistic regression to the training set.

         #Fitting Logistic Regression to the Training set

         from sklearn.linear_model import LogisticRegression
         classiffier = LogisticRegression(random_state = 0)
         classiffier.fit(x_train, y_train)


         #X_test contains the 2 principal components that were extracted and
         #predict the test set result.

Out[13]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
                   penalty='l2', random_state=0, solver='liblinear', tol=0.0001,
                   verbose=0, warm_start=False)
```

*Figure 8: Performing Logistic Regression to the training set*

The predicted test set result for X_test maintains the 2 principal components that were selected.

```
In [14]: #Predicting the Test set results

         y_pred = classiffier.predict(x_test)
```

*Figure 9: Predicting the test set result*

Confusion matrix is used to evaluate the model.

```
         #Making the confusion matrix
         from sklearn.metrics import confusion_matrix
         cm = confusion_matrix(y_test, y_pred)

In [15]: #Making the cufffusion matrix

         cm

Out[15]: array([[14,  0,  0],
                [ 1, 15,  0],
                [ 0,  0,  6]], dtype=int64)
```

*Figure 10: Confusion matrix - Evaluating the performance of the model*

Convert result of figure10 into the table1.

| n=36 | Predicted group A | Predicted group B | Predicted group C |
|---|---|---|---|
| Group A | 14 | 0 | 0 |
| Group B | 1 | 15 | 0 |
| Group C | 0 | 0 | 6 |

*Table 1: Prediction table*

Table 1 is the constructed confusion matrix showing a diagonal representation of correct predictions. The outcome is 14 cases of group A, 15 cases of group B, and 6 cases of group C. However, an incorrect prediction is observed, where the real outcome was group A but it was predicted to be group B.

There is 35 correct predictions divided by the total number of 36 cases providing us about 97.2% accuracy. Visualization of training result in Figure 11.

```
#class label color
for i, j in enumerate(np.unique(y_set)):
    plt.scatter(x_set[y_set == j, 0], x_set[y_set == j, 1], edgecolor= "black",
                c = ListedColormap(("gray", "white", "green")) (i), label = j)

plt.title("Logistic Regression (Training Dataset)")
plt.xlabel("Principal Component 1")
plt.ylabel("Principal Component 2")
plt.legend()
plt.show
```

```
Out[47]: <function matplotlib.pyplot.show(*args, **kw)>
```

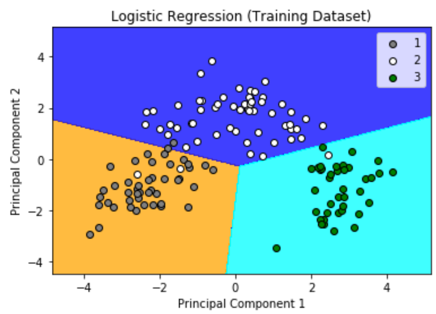


*Figure 11: Visualizing Training result*

In the visualization in Figure 11, shows the predicted regions

⇒ Orange = Group A
⇒ Blue = Group B
⇒ Aqua = Group C

The tiny circles are the main observations in the wine dataset

⇒ Gray = Group A
⇒ White = Group B
⇒ Green = Group C

Visualization of our test result is of utmost importance to be consistent with evaluated confusion matrix result.

 

```
#class label color
for i, j in enumerate(np.unique(y_set)):
    plt.scatter(x_set[y_set == j, 0], x_set[y_set == j, 1], edgecolor= "black",
                c = ListedColormap(("gray", "white", "green")) (i), label = j)

plt.title("Logistic Regression (Test set)")
plt.xlabel("Principal Component 1")
plt.ylabel("Principal Component 2")
plt.legend()
plt.show
```

```
Out[48]: <function matplotlib.pyplot.show(*args, **kw)>
```

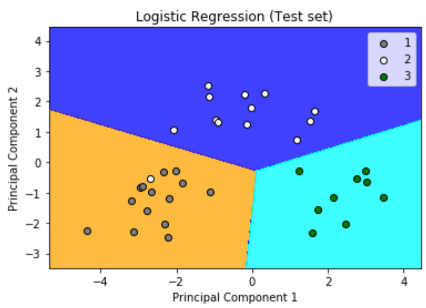


*Figure 12: Visualizing Test result*

An obvious white circle predicted in the yellow region which is meant for group B. PCA obtained a 97% accuracy.

## 1.8 Testing and Training for LDA

Since LDA is supervised technique, dependent variable ‘y_train’ is included. Fitting and classification model is done after the application of LDA.

```
#We apply LDA before fitting the classification model to the training set.
#One thing to note is because LDA is a supervised technique, we need to inclu

#The rest of what we do in this section should now be more familiar.


#Applying LDA

from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA
lda = LDA(n_components = 2)

#In contrast to PCA, we also need to include the y_train for the fitting_trar
x_train = lda.fit_transform(x_train, y_train)
x_test = lda.transform(x_test)
```

*Figure 13: Applying LDA*

```
#Fitting Logistic Regression tot the training set

from sklearn.linear_model import LogisticRegression
Classiffier = LogisticRegression(random_state = 0)
classiffier.fit(x_train, y_train)
```

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
          penalty='l2', random_state=0, solver='liblinear', tol=0.0001,
          verbose=0, warm_start=False)
```

```
#Predicting the Test set results

y_pred = classiffier.predict(x_test)
```

*Figure 14: Fitting Logistic Regression, and Classification*

```
# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
cm
```

```
array([[14,  0,  0],
       [ 0, 16,  0],
       [ 0,  0,  6]])
```

*Figure 15: Confusion matrix - Evaluating the performance of the model*

| n=36 | Predicted group A | Predicted group B | Predicted group C |
|---|---|---|---|
| Group A | 14 | 0 | 0 |
| Group B | 0 | 16 | 0 |
| Group C | 0 | 0 | 6 |

*Table 2: LDA Prediction table*

Table 2 shows correct predictions of 14 cases of group A, 16 cases of group B, 6 cases of group C and a LDA obtained a 100% accuracy.

```
#class label color
for i, j in enumerate(np.unique(y_set)):
    plt.scatter(x_set[y_set == j, 0], x_set[y_set == j, 1], edgecolor= "black",
                c = ListedColormap(("gray", "white", "green")) (i), label = j)

plt.title("Logistic Regression (Training Dataset)")
plt.xlabel("Linear Discriminant 1")
plt.ylabel("Linear Discriminant 2")
plt.legend(loc = "lower left")
plt.show
```

```
Out[52]: <function matplotlib.pyplot.show(*args, **kw)>
```

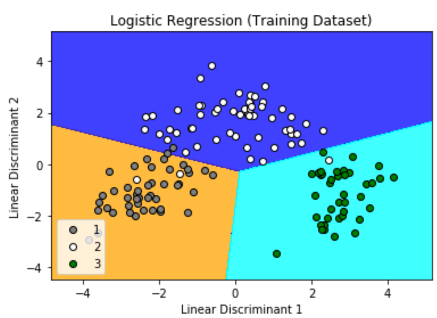


*Figure 16: Visualizing Training result*

```
#class label color
for i, j in enumerate(np.unique(y_set)):
    plt.scatter(x_set[y_set == j, 0], x_set[y_set == j, 1], edgecolor= "black",
                c = ListedColormap(("gray", "white", "green")) (i), label = j)

plt.title("Logistic Regression (Test Dataset)")
plt.xlabel("Linear Discriminant 1")
plt.ylabel("Linear Discriminant 2")
plt.legend(loc = "lower left")
plt.show
```

```
Out[53]: <function matplotlib.pyplot.show(*args, **kw)>
```

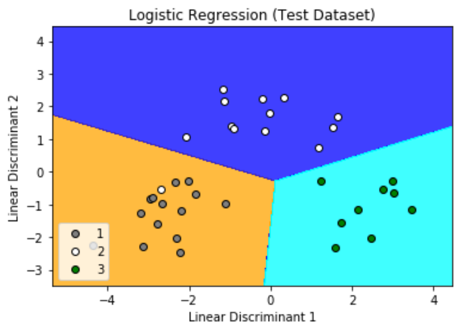


*Figure 17: Visualizing Test result*

## 1.9 Conclusion

PCA and LDA dimensionality reduction techniques provides avenue to select variables that possesses vital information for efficient classification according to type of chemical component in a region. In this case, PCA and LDA proved useful in reduction, classification and prediction. Our study showed a 97.2% close accuracy from the PCA and a 100% accuracy prediction from LDA which was done with the help of python modules.

# REFERENCES

Donoho DL, Elad M. Optimally sparse representation in general (nonorthogonal) dictionaries via 1 minimization. Proc Natl Acad Sci. 2013;100(5):2197–202.

Kondziolka Benjamin T C, Lunsford LD, Silverman J. Development, implementation, and use of a local and global clinical registry for neurosurgery. Big Data. 2015;3(2):80–9.

DongGuo H, Zhang L, WeiZhu L. Earth observation big data for climate change research. Adv Clim Change Res. 2015;6(2):108–17.

Efron B, Hastie T, Johnstone I, Tibshirani R. Least angle regression. Ann Stat. 2003;32:407451.

Laney D. 3D Data management: controlling data volume, velocity and variety. 2001.

H. B. Borges and J. C. Nievola, ``Comparing the dimensionality reduction methods in gene expression databases,'' Expert Syst. Appl., vol. 39, no. 12, pp. 10780_10795, 2012.

K. C. Tan, E. J. Teoh, Q. Yu, and K. C. Goh, ``A hybrid evolutionary algorithm for attribute selection in data mining,'' Expert Syst. Appl., vol. 36, no. 4, pp. 8616_8630, 2009.

F. Artoni, A. Delorme, and S. Makeig, ``Applying dimension reduction to EEG data by principal component analysis reduces the quality of its subsequent independent component decomposition,'' NeuroImage, vol. 175, pp. 176_187, Jul. 2018.

I. T. Jolliffe, Principal Component Analysis. New York, NY, USA: Springer-Verlag, 2002.

 

A. Balodi, M. L. Dewal, R. S. Anand, and A. Rawat, ``Texture based classi_cation of the severity of mitral regurgitation," Comput. Biol. Med.,vol. 73, pp. 157_164, Jun. 2016.

H. Alshamlan, G. Badr, and Y. Alohali, ``mRMR-ABC: A hybrid gene selection algorithm for cancer classi_cation using microarray gene expression proling," BioMed. Res. Int., vol. 2015, pp. 1_15, Mar. 2015.

X. Yan and M. Jia, ``Intelligent fault diagnosis of rotating machinery using improved multiscale dispersion entropy and mRMR feature selection," Knowl.-Based Syst., vol. 163, pp. 450_471, Jan. 2019.

A. E. Akadi, A. Amine, A. E. Ouardighi, and D. Aboutajdine, ``Atwo-stage gene selection scheme utilizing MRMR _lter and GA wrapper," Knowl. Inf. Syst., vol. 26, no. 3, pp. 487_500, 2011.

J. T. Souza, ``Methods of attribute selection and principal component analysis: A comparative study," Ph.D dissertation, Federal Technol. Univ.-Parana, Ponta Grossa, Brazil, 2017, p. 73. [Online]. Available: http://repositorio.utfpr.edu.br/jspui/handle/1/2387

A. Onan, ``A fuzzy-rough nearest neighbor classifier combined with consistency-based subset evaluation and instance selection for automated diagnosis of breast cancer," Expert Syst. Appl., vol. 42, no. 20, pp. 6844_6852, 2015.

Akash Desarda (2018). Retrieved from https://towardsdatascience.com/getting-data-ready-for-modelling-feature-engineering-feature-selection-dimension-reduction-39dfa267b95a, on 15 September 2019.

Akash Desarda. Getting Data ready for modelling: Feature engineering, Feature Selection, Dimension Reduction (Part two)

Ashwini Kumar Pal. Understanding Dimension Reduction with Principal Component Analysis (PCA) (11 DECEMBER 2017). Retrieved from https://blog.paperspace.com/dimension-reduction-with-principal-component-analysis/, 15 September 2019.

Claire. Powerful Dimensionality Reduction Techniques For Successful Analysis (2019). https://www.artificiallyintelligentclaire.com/dimensionality-reduction/, retrieved 15 September 2019.

Dataflair Team. Machine Learning Algorithms Tutorial – Which ML Algorithm is Best? (2018). Retrieved from https://data-flair.training/blogs/machine-learning-algorithms/, on 15 September 2019.

Deepai. Curse of Dimensionality. Retrieved from (2017). https://deepai.org/machine-learning-glossary-and-terms/curse-of-dimensionality, on 15 September 2019.

Elior Cohen. Reducing Dimensionality from Dimensionality Reduction Techniques (2017). Retrieved from https://towardsdatascience.com/reducing-dimensionality-from-dimensionality-reduction-techniques-f658aec24dfe, on 15 September 2019.

Falguni Mukherjee. Linear Algebra for Machine Learning Part 6— Dimensionality Reduction and PCA (2018). Retrieved from https://medium.com/@falgunimukherjee/linear-algebra-for-machine-learning-part-6-dimensionality-reduction-and-pca-5811d42339ec, on 15 September 2019.

François Cartier. What is the difference between a 2D data model and a 3D data model?(Aug 4, 2017) Retrived from https://www.quora.com/What-is-the-difference-between-a-2D-data-model-and-a-3D-data-model, 15 September 2019.

Hussein Abdullatif. Dimensionality Reduction For Dummies — Part 3: Connect The Dots Introduction to Dimensionality Reduction (2013). Retrieved from https://www.geeksforgeeks.org/dimensionality-reduction/, on 15 September 2019.

 

Judy T Raj. A beginner's guide to dimensionality reduction in Machine Learning(2019). Retrieved from https://towardsdatascience.com/dimensionality-reduction-for-machine-learning-80a46c2ebb7e, on 15 September 2019.

Khaing Win. How to Overcome the Curse of Dimensionality: DATA SCIENCE(2018). Retrieved from https://byteacademy.co/blog/overcome-dimensionailty-machinelearning, on 15 September 2019.

Margaret Rouse. Dimensionality reduction (2018). Retrieved from https://whatis.techtarget.com/definition/dimensionality-reduction, on 15 September 2019.

Marin Vlastelica Pogančić. On the Curse of Dimensionality (2019). Retrieved from https://towardsdatascience.com/on-the-curse-of-dimensionality-b91a3a51268, on 15 September 2019.

Mohit Deshpande. Python Machine Learning : Dimensionality Reduction (2017), retrieved from https://pythonmachinelearning.pro/dimensionality-reduction/, on 15 September 2019.

Pulkit Sharma. The Ultimate Guide to 12 Dimensionality Reduction Techniques (with Python codes) (2018). Retrieved from https://www.analyticsvidhya.com/blog/2018/08/dimensionality-reduction-techniques-python/, on 15 September 2019.

Rinu Gour. Dimensionality Reduction in Machine Learning (2019). Retrieved from https://medium.com/@rinu.gour123/dimensionality-reduction-in-machine-learning-dad03dd46a9e, on 15 September 2019.

S. Bleisch..et al. Connected 2d And 3d Visualizations For The Interactive Exploration Of Spatial Information (2017).

I.T. Jolliffe, Prinicipal Component Analysis, 2nd Edition, Springer series in statistics 2002, page 1-3.

Lindsay I Smith, A tutorial on Principal Components Analysis, February 26, 2002, page 2-8.

Sanghyun Paek, et al..Dimensionality Engineering of hybrid perovskites for photovoltaics and optoelectronic applications (2019).

Steve Dubel…et al. 2D and 3D Presentation of Spatial Data: A Systematic Review (2014). Retrieved from http://blogs.evergreen.edu/vistas/files/2015/02/dubel-topost-2dvs3d-visieeevis2014_submission_3.pdf, on 15 September 2019.

Sumithra V.S, et al. A Review of Various Linear and Non Linear Dimensionality Reduction Techniques (2015).

Sunil Ray. Beginners Guide To Learn Dimension Reduction Techniques 2015. Retrieved from https://www.analyticsvidhya.com/blog/2015/07/dimension-reduction-methods/, on 15 September 2019.

Tony Yiu. The Curse of Dimensionality (2019). Retrieved from https://towardsdatascience.com/the-curse-of-dimensionality-50dc6e49aa1e, on 15 September 2019.

Wei-Lun Chao. Dimensionality Reduction (2011)

XuechuanWang. Pattern Recognition: Feature extraction and dimensionality reduction algorithms and their applications in vowel recognition (October 2003).

Josua Krause, et al. SeekAView: An Intelligent Dimensionality Reduction Strategy for Navigating High-Dimensional Data Spaces (2016).

 

Rosaria Silipo, et al. 3 New Techniques for Data-Dimensionality Reduction in Machine Learning (2019). Retrieved from https://thenewstack.io/3-new-techniques-for-data-dimensionality-reduction-in-machine-learning/, on 16 September 2019.

Shiri Margel, et al. Clustering and Dimensionality Reduction: Understanding the “Magic” Behind Machine Learning (2017). Retrived from https://www.imperva.com/blog/clustering-and-dimensionality-reduction-understanding-the-magic-behind-machine-learning/, on 16 September.

Pinderjeet Kaur (2012), ‘Implementation of Principle Component Analysis with Fuzzy Annotation for CAD Jewellery Images’, Int. J. of Emerging Trends & Technology in Computer Science,Vol. 1,Issue 4. ISSN 2278-6856.

Arunasakthi. K and Kamatchipriya. L(2014), ‘A Review On Linear And Non-Linear Dimensionality Reduction Techniques’, Machine Learning And Applications: An Int. J. (Mlaij), Vol.1, No.1, Pp.65-76.

Julie M. David, Kannan Balakrishnan, (2014), Learning Disability Prediction Tool using ANN and ANFIS, International Journal of Soft Computing, Springer Verlag Berlin Heidelberg, ISSN 1432-7643 (online), ISSN 1433-7479 (print), DOI: 10.1007/s00500-013-1129-0, Vol. 18, Issue 6, pp 1093-1112

Julie M. David, Kannan Balakrishnan, (2012), Attribute Reduction and Missing Value Imputing with ANN: Prediction of Learning Disabilities, International Journal of Neural Computing & Applications, Springer-Verlag London Limited, DOI: 10.1007/s00521-011-0619, Vol. 21, Issue 7, pp 1757-1763

Kresimir Delac, Mislav Grgic and Sonja Grgic (2006), ‘Independent Comparative Study of PCA, ICA, and LDA on the FERET Data Set’, Wiley Periodicals, Inc.

Dr. H.B.Kekre, Sudeep D. Thepade and Akshay Maloo (2010), ‘CBIR Feature Vector Dimension Reduction with Eigenvectors of Covariance Matrix using Row, Column and Diagonal Mean Sequences’, Int. J. of Computer Applications (0975 – 8887),Vol. 3, No.12.

Sebastian Raschka (2014), ’Linear Discriminant Analysis bit by bit’.

 